Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

AUTOMATIC CONTROL LABORATORY
Prof. John Lygeros

Semester Thesis

Integration of a Walking Engine on a Bipedal Robot for RoboCup

Autumn Term 2019

Supervised by:
Alexandros Tanzanakis
Benjamin Flamm

Author:
Timo Roth

Declaration of Originality

I hereby confirm that I am the sole author of the written work here enclosed and that I have compiled it in my own words. Parts excepted are corrections of form and content by the supervis

Integration of a Walking Engine on a Bipedal Robot for RoboCup

Author(s)

Timo	Roth

Student supervisor(s)

Alexandros	Tanzanakis
Benjamin	Flamm

Supervising lecturer

John	Lygeros

With my signature I confirm that

- I have committed none of the forms of plagiarism described in the 'Citation etiquette' information sheet.
- I have documented all methods, data and processes truthfully.
- I have not manipulated any data.
- I have mentioned all persons who were significant facilitators of the work.

I am aware that the work may be screened electronically for plagiarism

Zurich, the 14th of February 2020

Place and date

Signature

Contents

Abstract

The ETH nomadZ robotic soccer team currently uses an outdated walking engine, which performs poorly compared to the walking engines used by competing teams. The limited capabilities also reduce the possible available team strategies. A promising walking engine has been integrated into the software framework, and compared with the old engine in terms of walking stability and speed. The new walking engine exhibits a significant improvement in both respects. Finally, several contributions are made to foot trajectory and walking controller parameters, which result in additional improvement in the stability and speed.

Acknowledgements

I want to thank the IfA and Prof. Dr. J. Lygeros for giving me this unique and interesting opportunity. My experience with the RoboCup NomadZ team gave me the chance to be part of a large and fascinating project. I learned a lot during this semester thesis and I will certainly benefit from it in the future.
I want to thank all the TA from Nomadz, Matthias Bräm, Lucia Liu and Koen Wolters for their application in the team and always being willing to help. Especially Koen, who was a really big help regarding the implementation and all the coding issues which came with it.
Finally, I'd like to thank my two supervisor, Benjamin Flamm and Alexandros Tanzanakis. They have always been a great support and they led me through this project very competently, with implication and encouragement.

Chapter 1

Introduction

This thesis was conduct as a contribution to the ETHZ RoboCup team project, Nomadz. The team is competing in the RoboCup standard platform where teams of 5 NAO robots are playing soccer and facing each other. The robots have to operate fully autonomously during the game which represent many interesting challenge in various fields like computer vision for environment recognition and localization, behavior control to elaborate strategies as a team and motion control to ensure stable and efficient locomotion. This thesis deal with this third domains, specifically the walking engine.

The motivation for this project was brought by the observation of the teams during the last tournament where the robots were outperform by the other teams, especially in term of speed. The used walking engine was indeed outdated and was the limitation factor for the behavior of the robots. It was therefore needed to improve the locomotion performances of the robots to allow them to execute more advanced strategies.

The goal of this thesis was therefore to improve the walking engine used by the Nomadz team. As the Robocup is an open platform, the team are communicating sharing their codes. The objective was to investigate other implementation, to understand their underlying concepts and their compatibility with the current framework. After this first phase the most promising walking engine was implemented on the NAO robots. And, subsequently, the previous and the new walking engine were tested to compare their performances. Finally, a the last phase was dedicated to identify some weakness of the new walking engine and to propose some solutions to try to further improve it.

Chapter 2

Theoretical Principle of the Walking Engines

This first chapter, is dedicated to present the underlying theoretical concept of the walking engines. Firstly, the previously used walking engine is described and secondly, the new one is introduced.

2.1 Dortmund Walking Engine by team NaoDevils

This walking engine was first developed by the team NaoDevils from the University of Dortmund and is named "Dortmund walking engine" it has been used until today by the Nomadz team. A complete description can be found in the NaoDevils' team report. [1]

2.1.1 Zero Moment Point (ZMP)

The Zero Moment Point is a commonly used factor to asses stability in bipedal locomotion. The interaction between the foot and the ground cannot be controlled directly and generate temporary passive DOF but therefore there is need for an indirect way to control those. This can be done by ensuring the appropriate dynamics of the mechanism above the foot. Thus, the overall indicator of the mechanism behavior is the point where the influence of all forces acting on the mechanism can be replaced by one single force. [2]
The fig. 2.1 summarize the interaction forces on the support foot of a biped robot. The influence of all the body of the robot is modelize as a reaction force $\mathbf{F}_A$ and a reaction momentum $\mathbf{M}_A$ applied at the ankle joint. The reaction on point on the ground consist of a force $\mathbf{R}(R_X, R_Y, R_Z)$ and a momentum $\mathbf{M}(M_X, M_Y, M_Z)$.
It is then shown in [2] that the point on the ground where a single reaction force is acting is then the point where:

$$
\begin{aligned}
M_X &= 0 \\
M_Y &= 0
\end{aligned}
$$

Which is named Zero Moment point.
There is then two cases to distinguish.

1. The ZMP lies inside of the support foot area and the mechanism is stable. In this case the ZMP is also the center of pressure.

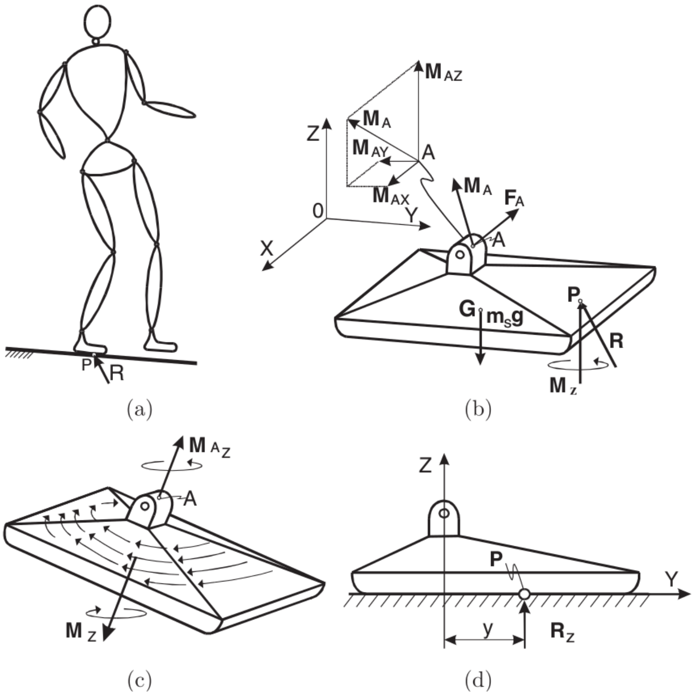

Figure 2.1: Forces acting on the foot and on the ground for a biped mechanism [2]

2. The ZMP lies outside of the support foot area and is therefore called fictitious ZMP. The reaction force is then acting on the point of the edge of the foot which is the closest to this fictitious ZMP. Therefore a there will be a remaining momentum acting on the foot which will make the robot rotate around the edge of the support foot.

A good condition for stability is then to keep the zero moment point within the support polygon of the mechanism.

2.1.2 Flexible Linear Inverted Pendulum

It is common to model a bipedal mechanism by an inverted pendulum. The team Nao Devils uses a more sophisticated model than a simple pendulum and developed a flexible linear inverted pendulum model [3]. In this model the force acting on the system is not directly applied on the cart where the pendulum is fixed but on another cart which is connected to the main cart by a spring damper. This flexible part can be seen as a representation of the flexible parts in between the foot and the center of mass of the robot, e.g. the gears in the joints.
the fig. 2.2 show a 2D schematic of the FLIP model where p is the ZMP position of the first cart, c_1 the position of the pendulum center of mass, and z_h the height of the pendulum center of mass.
From this model we can derive an equation for the position of the ZMP, which is shown in [3].

$$p \;=\; c_1 \;+\; \frac{z_h}{gm_1}\,(b\dot{d}_1 + kd_1) \tag{2.1}$$

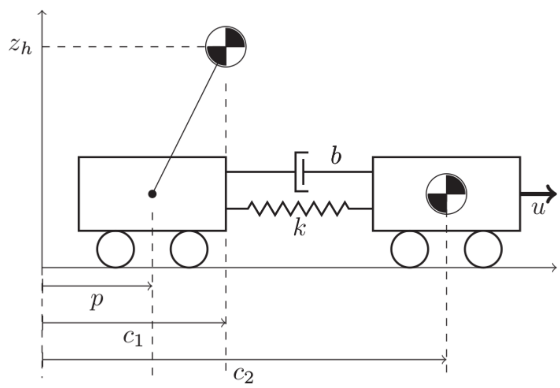

Figure 2.2: Schematic of the FLIP model[3]

2.1.3 LQR Regulator

From the FLIP model a state space representation of it is derived and an observer based Linear Quadratic Regulator is computed.

This controller take as input a reference ZMP trajectory and provides as output a target position for the center of mass of the robot.

The complete derivation of this controller is done in [4].

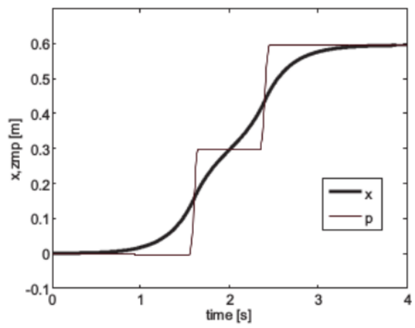

Figure 2.3: Poisition of the center of mass following the reference ZMP through the LQR regulator

2.1.4 Block Diagram

A summary of the Dortmund walking engine principle is shown in this block diagram. The target speed command (lateral, forward and angular) for the robot is given as input. Then a foot steps pattern is generated. From this footstep pattern the swinging leg controller is providing the necessary position of the foot to follow those footsteps. In parallel, The ZMP generator compute the reference ZMP which is the input for the LQR controller. The outputs from the LQR controller and the swinging leg controller are combined and the desired joint values for the robot are computed using inverse Kinematic. The robot provide the real center of mass thank to its joints position and the estimation of the real ZMP using it's inertial sensors.

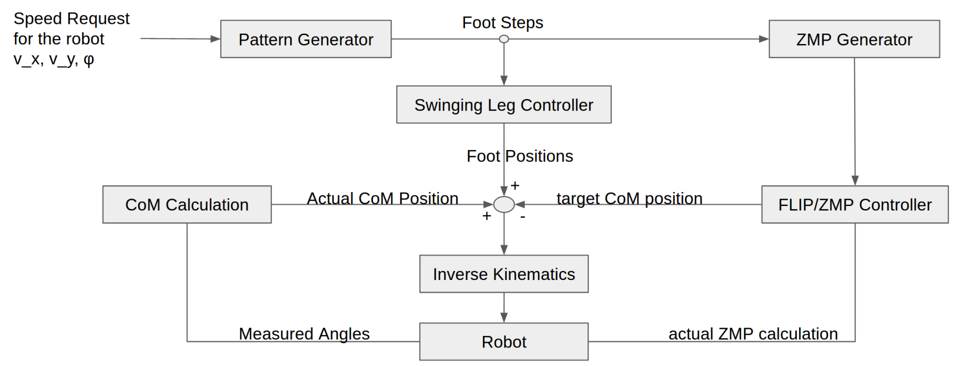

Figure 2.4: Block Diagram of the Dortmund Walking Engine system

2.2 Walk2014 by team rUNSWift

This walking engine was first developed by the team rUNSWift from the University of South Wales and is named "Walk2014". The principle is fully detailed in the paper from Berhard Hengst [5].
It's a much simpler model which take advantage of the natural behavior of the Nao robot.
To describe and control the walking engine, the dynamics of the robot is separate into two planes:

1. The lateral plane
2. The Longitudinal

2.2.1 Lateral Plane

On the lateral plane, the walking engine uses the natural oscillation of the robot to generate the steps. This illustrated in fig. 2.5 where the robot is modelised as an inverted pendulum whose pivot points on the edge of the feet change one after each other as the robot rock from side to side, lifting one foot after the other.

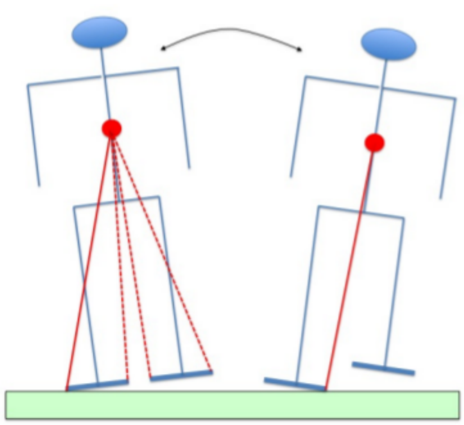

Figure 2.5: Illustration of the robot lateral rock[5]

The gravitational force acting on the bob of the pendulum will push the robot to balance back to the swinging foot while the pivot is located on the extern part of the support foot as it is illustrated in fig. 2.6 which will maintain the robot oscillation.

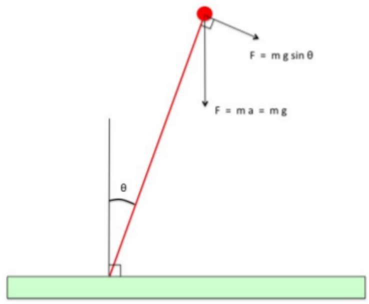

Figure 2.6: The gravitational force on the bob of an inverted pendulum [5]

The oscillation is automatically initiated by the reaction forces generated by the lifting of one foot.

2.2.2 Longitudinal Plane

On the longitudinal plane, the locomotion of the robot is generated by moving the support foot backward and the swinging foot forward relatively to the center of mass of the robot.

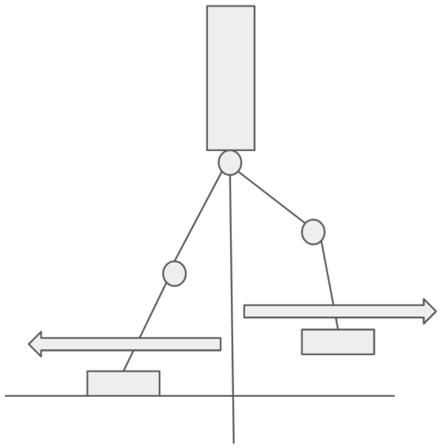

Figure 2.7: Illustration of the generation of the longitudinal displacement of both feet to make the robot move forward.

2.2.3 Feedbacks

Lateral feedback

The open-loop on the lateral plane assume the oscillation period to be constant. But any disturbance would modify this periode and could induce instability. Therefore a feedback is needed to detect the switch between the support and the swinging foot.

This can be done by using the pressure sensors under the feet of the Nao robot. There is 8 of them (for per foot). By measuring their values it is possible to compute the center of pressure (COP). The values of the COP are positive for the right side of the robot and negative for the left side with the origin in the middle, between the feet. The end of a step and the beginning of a new one can therefore be triggered by the zero crossing of the COP.

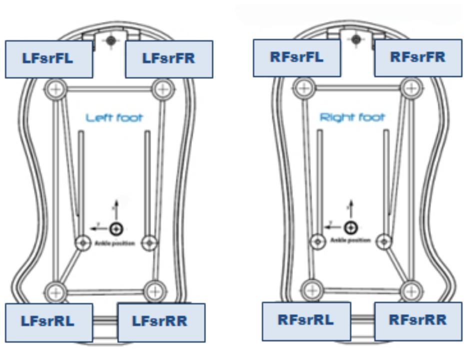

Figure 2.8: foot sensor localisation under both foot of the Nao Robot

Longitudinal feedback

The movement of the robot would inevitably induce disturbances. Without any feedback to control the posture of the robot there is no way to avoid those disturbances and to ensure the robot to keep equilibrium.
The Nao rabot are equipped with some inertial sensors, located in torso, a 3 axes accelerometer and a 3 axes gyroscope. The goal of the feedback is to keep the torso as straight as possible. The solution to achieve this is to measure the value of the y gyroscope, which is the torso pitch. The values from this sensor is the angular speed of the torso in [rad/s]. This value is multiplied by a gain and directly feeded back to the ankle pitch. By this way, the ankle pitch of the support foot is modified to counter the leaning of the torso and to keep it as straight as possible.

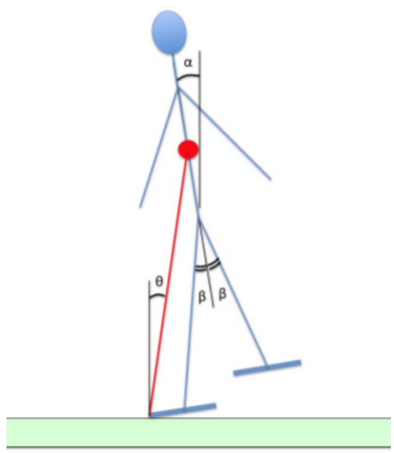

Figure 2.9: Illustration for the feedback principle on the longitudinal plane. The angular speed $\dot{\alpha}$ is measured and used to correct the ankle pitch θ

2.2.4 Block Diagram

A summary of the Walk2014 principle is shown in this block diagram. The target speed command (lateral, forward and angular) for the robot is given as input. After that the parameter for the next step are computed, e.g. the goal position for the feet. After this, the displacement of the feet for the next time step (the joint values are actualized every 0.01[s]) is computed following predefined trajectory (see section 5.2). Then the Ankle pitch is corrected with the feedback from the gyroscope and finally the joint values are computed using inverse kinematic to reach the desired position of the feet. If a new step is detected by the center of pressure computation the next step parameters are computed, otherwise we simply move to the next time step.

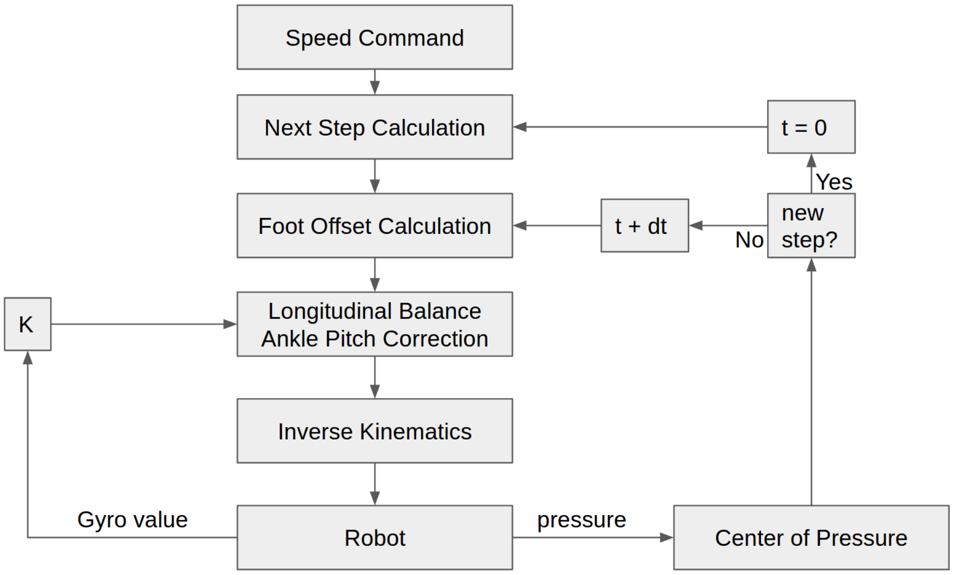

Figure 2.10: Block Diagram of the Walk2014 system

Chapter 3

Implementation

One big part of this semester has been to implement the new walking engine in the framework from Nomadz.

3.1 Motivation

Although the *Walk2014* used a much more simpler approach than the *Dortmund Walking Engine* and involves much less complex modelling and control computation, the experience of the member of the team during the last tournaments was telling that the teams who were using this walking engine were much more efficient than Nomadz.
That's why it was decided to try to make the *Walk2014* work on the robots.

3.2 Integration of the new Walking Engine in the Nomadz Framework

For the integration of this new walking engine in the Nomadz framework, it was choosen to take the implementation made by the team B-human which is also using it[6] . The Nomadz framework phase firstly based on the B-Human framework at the beginning of the team. Therefore, their code would be more compatible with the one from Nomadz.
Three modules where needed:

- **walk2014Generator** : This is the core of the walk2014 computation.
- **walkingEngine** : This is a wrapper for the walk generator. It take the data from the generator and publish the output of the walking engine in function of the motion request.
- **FootSupportProvider** : This modules take the datas from the foot sensors, compute the center of pressure and check if there is a new step to begin.

The fig. 3.1 shows a simplified representation of the newly imported modules in the Nomadz Framework.

Remaining Issues

Despite the similitudes between the frameworks, the implementation took a lot of time and rise a number of compilation and compatibility issues.

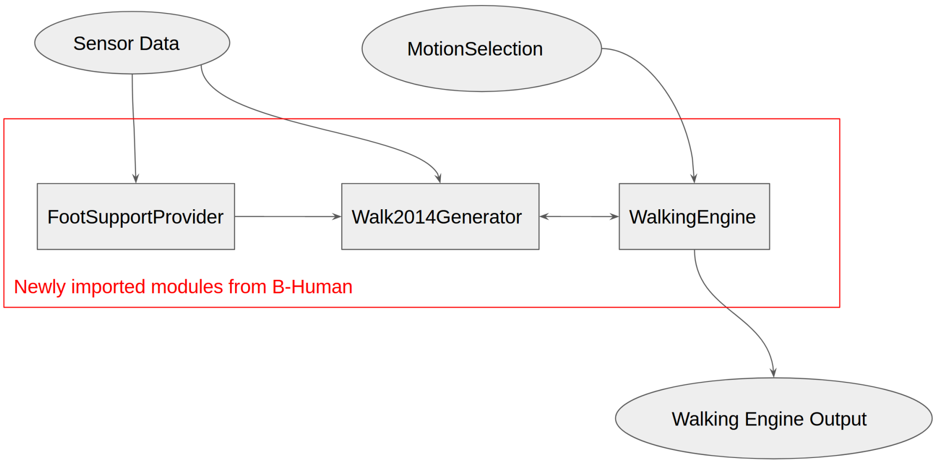

Figure 3.1: Illustration of the implemented modules in the Nomadz framework

Some of those issues are still remaining by the time of the writing of this report. The main three are listed here:

1. Switching to "stand" mode create a seg fault
2. Target mode for the walk is not working
3. Some joint values(hips roll, shoulder roll, wrist) are inversed

Hopefully they will be solved in the next weeks or documented in the gitlab of Nomadz.

Chapter 4

Comparison

Once the new walking engine was implemented. Some testing were conducted to find out the improvement it was bringing to the robot locomotion.
Two main things were important to test to have a good idea of the performances of the walking engine.

1. The capacity of the robot to evolve on a field with disturbances
2. The speed it can safely reach.

4.1 Experiments

4.1.1 Speed Comparison

The first experiment was meant to compare the speed at which both walking engine were able to move the robots.
The picture fig. 4.1 shows the setup that was used to do the experiment. The Robot was commanded to move at a certain speed and the time it needed to go through half of the field in the lab (approximately 4m) was measured.

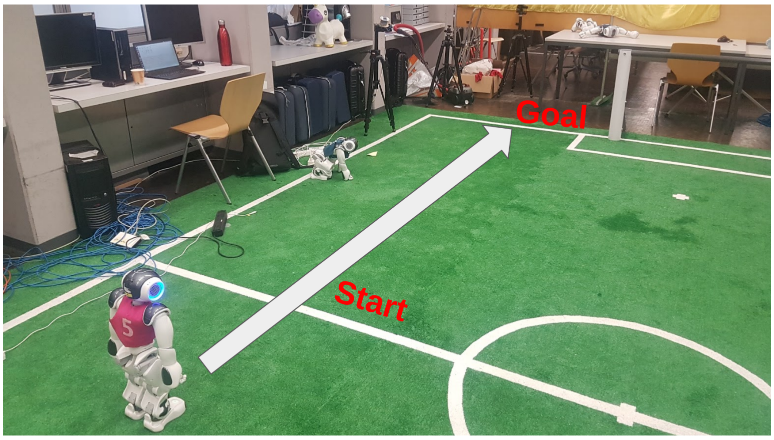

Figure 4.1: Description of the setup for the speed testing of the walking engines

4.1.2 Reaction to Disturbances

The second experiment was meant to compare the behavior of the robot in presence of disturbances while using both of the wlaking engine.

To create the disturbances some electric cables with different thickness were disposed on the floor and the robot was commanded to walk on them at a constant speed.

Figure 4.2: Ilustration of the setup for the reaction to disturbances testing of the walking engines

4.2 Results

4.2.1 Speed Comparison

The results of the speed comparison are shown in the two next tables. The first line is the command that was given to the robot, it has no particular unit but it's proportional to the actual speed of the robot. The second lines display the measured time.

command speed	180	180	180	200	200	200	220	220	220	230	230	230
time [s]	22.8	20.6	23.0	18.6	19.7	20.0	22.5	16.0	16.2	19.8	15.8	20.0

Figure 4.3: Table showing the results of the speed experiments with the *Dortmund walking engine*

command speed	200	230	230	230	250	250	275	275	275	280	280	290
time [s]	19.2	16.1	17.2	16.7	15.6	15.7	14.0	13.8	14.2	fall	15.0	fall

Figure 4.4: Table showing the results of the speed experiments with the *Walk2014*

Comments

- The command speed of 230 for the *Dortmund Walking Engine* was the max allowed speed by the program.
- The slow times for the *Dortmund Walking Engine* at commmand speed 220 and 230 can be explain by the fact that a higher speed induce bigger disturbances induced by the movements. The robot had to slow down or stop to find back is equilibrium and to start moving again.

- The maximal almost guaranteed speed for the *Dortmund Walking Engine* is around 16.0 [s] as for the *Walk2014* it can safely reach 14.0 [s] which represent an improvement of 12.5 [%] for the new walking engine
- If the speed become too big, the *Walk2014* is not able to catch up and fall.

4.2.2 Reaction to Disturbances

The results of this second experiment can be seen on video by following the two links.
video of the old walking engines:
`https://youtu.be/Hwyd0kHIGQ8`
video of the new walking engine:
`https://youtu.be/BkJ6yEp-ssU`

Comments

- The *Dortmund Walking Engine* faced the same problem as when it was moving close to maximal speed. He always has to stop to find is equilibrium back which prevent him from going through the obstacle efficiently. However, even though he is not able to keep walking efficiently, he is never falling.
- On the other hand the *Walk2014* was able to keep on walking even when it was stepping on the cable. This might be a benefit from the fact that the natural lateral balancing period of the robot is simply modified by the disturbances but this is easily overcome by the new step detection feedback.

4.2.3 Conclusion

The new walking engine bring some improvement. First it increase the speed capability of the robot by around 12.5 [%]. And second it react much better to small disturbances and he is able to adapt to them to keep on walking. One setback of the new walking engine is that instead of the old one, is that if begin to lose equilibrium, it can't find it back and will eventually fall down as it was seen with the high speed. The new walking engine seems nonetheless to be an improvement for the team. But there is still a lot of room for improvement which can be made and this is the topic of the next chapter.

Chapter 5

Contribution to the Walk2014 Walking Engine

During the comparison phase some weaknesses could be noticed while observing the robot moving with the new walking engine. In this chapter, those weaknesses are described and ideas of solution are proposed to improve the behavior of the robots which use this walking engine.

5.1 Limitation of the Walking Engine

The gait of the robot looked a little brutal and shaky, inducing some big oscillation of the torso. Moreover the robot had sometimes a tendency to stuck his feet in the ground while putting done is swinging leg, eventually making him fall forward and at high speed the tendency was for him to fall backward.
Three ideas were issued to try to improve those weaknesses.

1. Adding a retraction in the swinging foot trajectory. Indeed, this trajectory was not including one. Therefore the foot was finishing it's trajectory at 0 speed in the robot frame. Making the foot hit the ground at the speed of the robot,resulting in a shock.
2. Trying to add a feedback from the speed to the repartition of displacement of movement between the support and the swinging foot.
3. Tune the longitudinal feedback gain and try to find its optimum.

5.2 Swinging Leg Trajectory

To move forward the robot needs to push his support foot backward and it's swinging foot forward. This is done in the code by multiplying the distance at which the foot need to move at each time step by a function. This function is linear for the support foot and quadratic for the swinging foot. The quadratic function allow the swinging foot to be lifted at small speed, to move faster while it's up and to decelerate before landing on the ground. The problem of this function was that at the end it was finishing with a speed equal to zero in the robot frame, respectively the speed of the robot relatively to the ground. The difference of speed between the foo and the ground was then creating disturbances. The proposed solution to this issue is to shape the quadratic function for the swinging foot trajectory so the foot will hit the ground at zero speed relatively to it.

5.2.1 Trajectory shaping

The function describe the percentage of the distance in function of the percentage of the time of one step. From $t = 0\%$ to $t = 50\%$, $x_1(t)$ is a parabole with a positive curvature.

$$x_1(t) = a \cdot t^2 \tag{5.1}$$

And from $t = 50\%$ to $t = 100\%$, $x_2(t)$ is a parabole with a negative curvature.

$$x_2(t) = b \cdot t^2 + c \cdot t + d \tag{5.2}$$

Some constraints where used to identify the parameter $a, b, candd$:

$$\begin{cases} x_1(0.5) &= x_2(0.5) \\ \dot{x}_1(0.5) &= \dot{x}_2(0.5) \\ x_2(1) &= 1 \\ \dot{x}_2(1) &= -1 \end{cases}$$

The two first constraints are here to ensure continuity between x_1 and x_2. The 3rd constraint ensure that 100% of the distance is execute at the end of the step. And the last one specify the speed of the foot which should be $-(speed\ of\ the\ robot)$ at the end of the step.

Solving this linear system to find the parameters resulted in the new foot trajectory shown in fig. 5.1

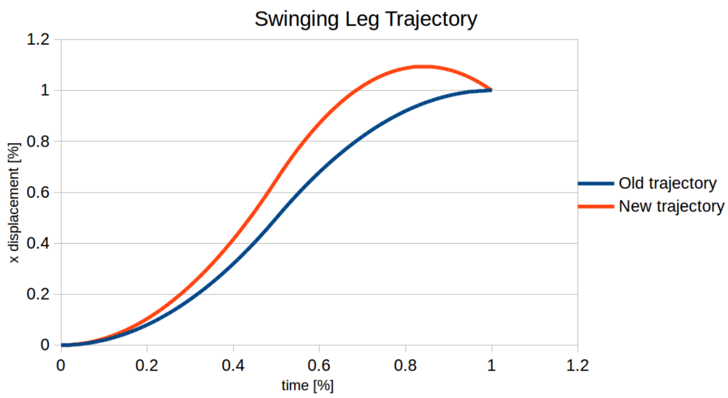

Figure 5.1: Trajectory of the swinging foot with $(oldtrajectory)$ and without $(newtrajectory)$ retraction

5.2.2 Results

Both trajectory were tested on the robots to show the differences. The qualitative result can be seen on the two videos.
link video without:
`https://youtu.be/wDzSSplip0s`
link video with:
`https://youtu.be/a8Npzk5boD4`

Secondly the same speed experiment as in chapter 4 was conducted with and without the foot retraction. The results can be seen in the following tables.

command speed	200	230	230	230	250	250	275	275	275	280	280	290
time [s]	19.2	16.1	17.2	16.7	15.6	15.7	14.0	13.8	14.2	fall	15.0	fall

Figure 5.2: Table showing the results of the speed experiments with the *Walk2014* and no foot retraction(same as in chapter 4)

command speed	230	230	230	250	250	250	275	275	275	300	300
time [s]	14.6	14.8	14.3	13.1	13.6	13.4	12.0	fall	11.8	fall	fall

Figure 5.3: Table showing the results of the speed experiments with the *Walk2014* and foot retraction

Comments

- The retraction of the swinging foot allow the robot to move more smoothly and reduce the amplitude of the oscillation of the torso while walking.
- Benefiting from this softer behaviour the robot was able to move faster and to safely reach time around 13[s] which is another 8% improvement.

5.3 Step Repartition Between Swinging and Support Foot

In the robot frame, the movement of the robot is creating by giving half of the displacement to the support foot moving backward and half of the swinging foot moving forward. This is illustrated on the left schematic in fig. 5.4. But, as the time for a step is dictated by the lateral rock of the robot, this time is fixed and the only solution to move faster is to do bigger step. This create a limitation factor for the speed as from a certain speed the swinging foot would move too much forward and land in an unstable region, making the robot lean backward and eventually fall. This is illustrated on the right picture of fig. 5.4.

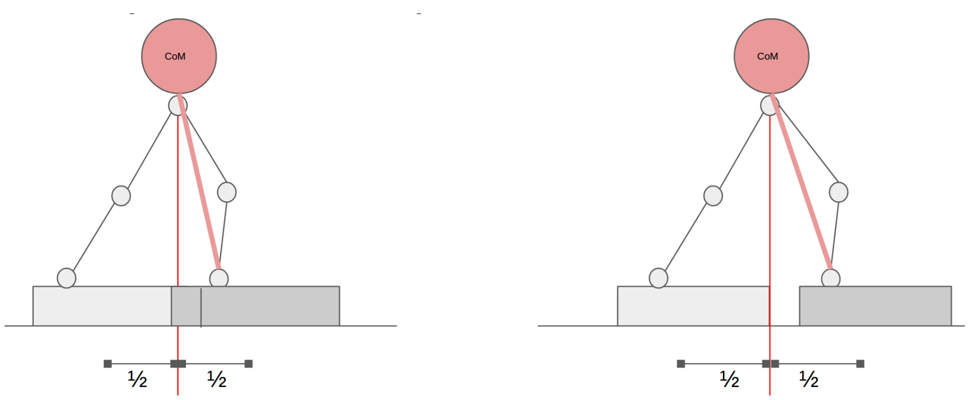

Figure 5.4: Illustration of the repartition of the distance executed by each of the foot

5.3.1 Idea of solution

The idea solution come first from a biomechanical intuition. Indeed when a human being is walking, the swinging foot is always landing really close to the projection of the center of gravity on the ground, no matter at which speed he's moving, it's even more clear for running.
Following this intuition, a solution might be to parametrize the repartition of displacement of the feet in function of the speed. As the speed increase more percentage of the displacement would be given to the support foot to ensure that the swinging foot always land in a stable region. The illustration of this idea is shown in fig. 5.5 where x is a parameter proportionnal to the speed of the robot.

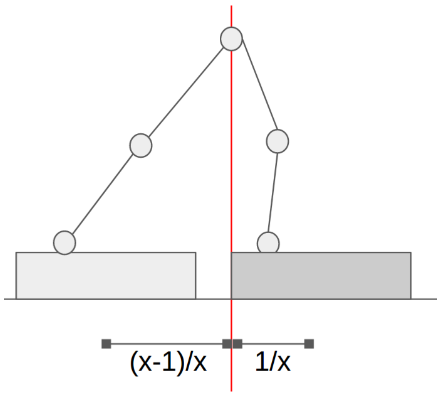

Figure 5.5: Idea of solution: a parametrize repartition of the displacement between ten feet in function of the speed of the robot

5.3.2 Results

This principle was tested on the robot and it influences the behavior of the robot and it's stability. However, the robot wasn't able to safely move faster. Instead of leaning backward, the support foot pushing too much backward made the robot lean forward, also resulting in falling. Nonetheless, the result is promising but need further development. For exemple by also including a feedback from the speed on the ankle pitch to push the torso to stay straight..

5.4 Longitudinal Feedback Tuning

The last contribution to improving the walking engine was trying to find an optimal gain for the longitudinal feedback.

5.4.1 Experiment Description

To conduct this experiment, the gyroscope value has been observed to determine which gain result in the smallest variation of this value.

5.4.2 Results

To vizualize the results they are shown as a praphic of the variance of the gyroscope value along the experiment (approximately 24 steps) in function of the gain.
The variance follow a partally parabolic curve with a minimum around a 0.06 gain. This is also illustrated in the next graph where the gyroscope value in function of

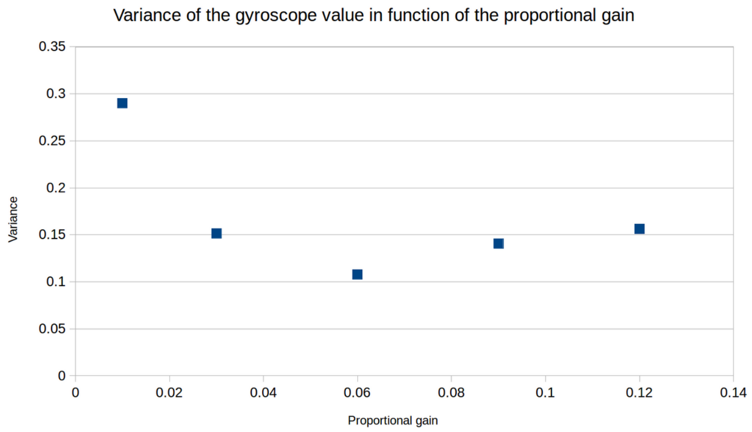

Figure 5.6: bla

the time are shown. It can be seen that the curve for the 0.06 gain has much smaller oscillations than the other curves.

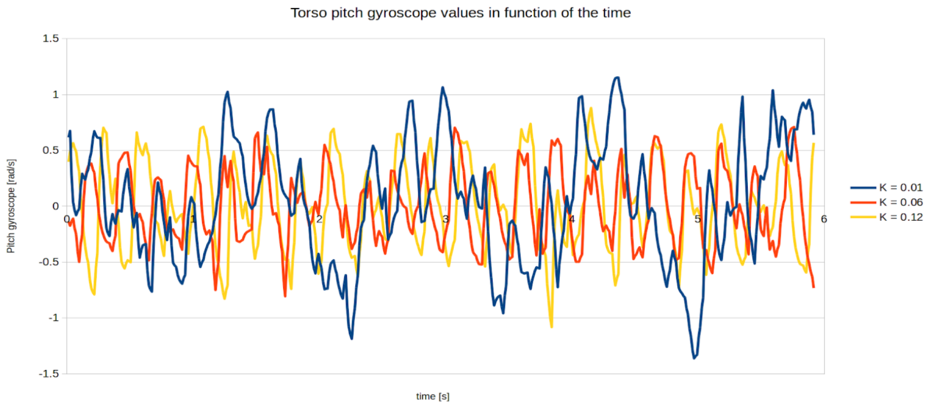

Figure 5.7: blabla

However, this gain alone cannot remove all the oscillations. The explanation for this could be that the robot is not a simple center of mass but as multiple joints and limb which influence it's reaction to the feedback.

Chapter 6

Conclusion

Achievement

The new walking engine was succesfully imported in the Nomadz framework. It has shown good result in the experiments, allowing the robot to move faster and to be able to keep on walking efficiently while being subject to disturbances such as cable on the floor. Some further improvement were made to the new walking engine to further improve the capabilities of the robot. The implementation of the foot retraction for the swinging leg was a significant improvement on the stability of the robot gait. And, it was shown that it was possible to find an optimal gain for the longitudinal balancing.

Limitation

The new walking engine is a much simpler model, which has its advantages but some disadvantages too. Having less control feedback paths don't allow the robot to recover from bigger disturbances and, ones it lost its equilibrium, it can't avoid to fall down.

Further work

Some possible way to keep improving the walking engine could be to add more parameter to the control feedbacks. For example adding some derivative and integral gains or allowing more joints (as the knee, the hips or even the arms) to be involve in the feedback loop. Some more complex control method as LQR regulator could also be tested in this walking engine. Finally, the ideas presented in the point 5.2 and 5.3 could be further developed to reduce the disturbances during the walk and to increase the speed capabilities of the robots.

Bibliography

[1] M. Hofmann, I. Schwarz, O. Urbann, and A. Larisch, "Nao Devils - Team Report 2018," 2018, https://github.com/NaoDevils/CodeRelease/blob/master/TeamReport2018.pdf.

[2] M. Vukobratović and B. Borovac, "Zero-moment point — thirty five years of its life," *International Journal of Humanoid Robotics*, vol. 1, no. 01, pp. 157–173, 2004.

[3] O. Urbann, I. Schwarz, and M. Hofmann, "Flexible Linear Inverted Pendulum Model for Cost-Effective Biped Robots," *IEEE-RAS International Conference on Humanoid Robots (humanoids)*, 2015.

[4] O. Urbann and S. Tasse, "Observer based biped walking control, a sensor fusion approach," *Autonomous Robots*, vol. 35, 07 2013.

[5] B. Hengst, "runswift walk2014 report," 2014, http://cgi.cse.unsw.edu.au/~robocup/2014ChampionTeamPaperReports/20140930-Bernhard.Hengst-Walk2014Report.pdf.

[6] T. Röfer, A. Baude, T. Laue, J. Blumenkamp, G. Felsch, J. Fiedler, J. Hasselbring, T. Hass, J. Oppermann, P. Reichenberg, S. Nicole, and D. Weiss, "B-Human - Team Report and Code Release 2019," 2019, https://github.com/bhuman/BHumanCodeRelease/blob/master/CodeRelease2019.pdf.