Taste of Research 2010-2011: Locomotion

Brock White

Supervisor: Bernhard Hengst

February 22, 2011

Contents

1 Introduction

This report will detail the changes made to locomotion throughout Taste of Research 2010-2011. Much of this will be covered in greater detail in the thesis I intend to write after robocup, however this report aims to detail a lot of the minors changes that have occurred to locomotion so that new team members may better understand the motion engine.

2 Related Work

The primary source and reference for my work over Taste of Research was the 2010 walk and report [2]. This details the 2010 Fast walk from which the new walk developed is based on. A review of this work can be found on the rUNSWift wiki in my Thesis A report (http://runswift.cse.unsw.edu.au/confluence/display/rc2011/Motion).

For related work in pendulum models and machine learning, my honours thesis will explore these works when it is released after robocup.

3 Method

3.1 New Motion Interface

The old motion interface required the user to set fields in three structs. One struct determined Head movement, one LEDs and the other Body movement. Another value was also set for actionType that is used to determine weather you are performing one of a number of actions including kicks, walks and getup routines.

In the new motion interface the parameters sent to the Body have changed. The parameters were previously forward, left, turn and power. If the actionType was set to WALK, the forward, left and turn parameters specified the desired velocity of the walk in those directions. If set to KICK, forward, left and turn were overloaded to specify the direction of the kick. In addition to this, the power parameter was used to determine the power of the kick. For more detail see the rUNSWift 2010 report [2].

Under the new interface; forward, left and turn have now been altered to represent the distance the walk is required to cover to reach its destination. The then being that motion determines what velocity to use to best reach its destination. A new speed parameter has been added to allow one to specify the maximum speed of the walk.

For kicks; forward, left and turn are no longer used. Instead we now have a new kickDirection and foot parameter. kickDirection is used to determine the direction we are to kick and is specified in radians. Foot is used to determine which foot to use. So for example, kickDirection = 0 with foot=RIGHT is a forward kick using the right foot.

3.2 Walk Architecture

As our walk engine improves, we will always be aiming for more robust human like gaits. These gaits will almost certainly involve some degree of step planning and body modeling to allow for this planning. The 2010 walk engine utilized basic versions of both of these features and integrated these directly into the walk engine.

To allow our code to move in this direction it was decided that for clarity the walk should be partitioned into distinct parts.

Planner This component has the responsibility of interpreting action commands and converting these into step requests. In the future we aim for this to involve some form of planning such that we remain balanced. For now this planning simply ensures that we do not change velocity too

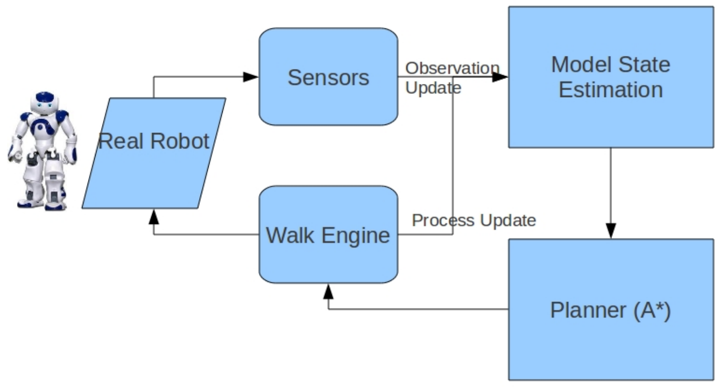

Figure 1: Relationship between the robot, its sensors and the new motion architecture.

rapidly.

BodyModel As the name suggests this component aims to model the body. For now this is just simple ZMP filtering for direct feedback balancing. In the future we are aiming for this to be a pendulum model. The Planner will use this BodyModel to make informed decisions about which steps to take.

WalkEngine This component is responsible for taking step requests and using these to create joint angles. It is also responsible for performing kicks as we have integrated kicks in the walk engine this year.

3.3 Step planner

The first version of the step planner was simply a direct extraction of the parameter scaling code form the 2010 walk. The idea is that we have input values for forward, left and turn. We then have internal real forward, left and turn values for the step planner. We then insure that we interpolate slowly towards the required parameters. By interpolating slowly we then ensure that no sudden centre of pressure changes occur and thus we usually remain balanced. The rate we interpolate determines the acceleration capabilities of the walk.

Only one major change occurred to that planning extracted from the 2010 walk. This was to add in the ability for discrete steps. To do this the step planner only changes its step requests between steps. This sounds natural but was not how the walk worked previously.

With this ability we are now able to choose to take single precise steps. This allows us to stop quickly when lining up with the ball. This new system is not without fault however as it often causes the centre of pressure to shift suddenly as our step size must change rapidly to stop suddenly. In the future this may be fixed via the use of more complex step planning using a pendulum body model.

Another smaller change was to only allow the walk to accelerate when it is operating at a desired step frequency. This was to avoid situations in which the walk starts turning or side stepping when the walk is first starting to ramp up (approximately 1 second). In the future we may reduce the time it takes to "ramp up the walk" by the use of coronal rock.

The plans for my thesis regarding the step planner are included at the end of this report (see 3.7).

3.4 Body Model

A basic body model already existed in the 2010 walk. This model was primarily a set of filters over the ZMP sensors (for centre of pressure) and accelerometers. These filtered results were then used to provide feedback based balance.

I will be attempting to implement a pendulum model in my thesis to replace or augment the existing filter. The initial plans of this are included at the end of this report (see 3.7).

3.5 Walk Engine

3.5.1 Parameterized Walk

To avoid slow incremental changes to the 2010 walk we decided to parameterize the 2011 walk. This entailed adding a series of variables to the walk such that we could alter the characteristics of the walk through these variables. The primary parameters added were:

CoronalRockAmplitude This effects the maximum rock angle of the body during the walk cycle. This simply changes the hip pitch and ankle pitch angles such that the robot rocks from side to side during the walk cycle. Currently set to **0 degrees**.

CoronalRockPhaseOffset This effects the phase offset of the coronal rock. The idea is that we thought this could be used to shift centre of mass to smooth out walk. This parameter still needs to be looked at more and is currently set to **0**.

KneeBend Degree of knee bend. Greater knee bend typically results in more stress on knee joint however it lowers Centre of mass allowing for easier balancing. Currently this is set to **15 degrees**.

LegTullip This effects the hipRoll angle such that the legs tilt towards the centre. This then effects the degree that the base of the feet are together. We think this can be used to allow for smoother steps. Currently we have this set to **1 degree**.

StepFrequency This effects the step frequency. Currently set to **0.5 seconds**. This is inter-lated with Coronal Rock Amplitude.

While these parameters have been hand tuned it would be nice to machine learn these in the future.

Algorithm 1 Smooth function for interpolation in kicks

```
float WalkEngine::interpolateSmooth(float start, float end, float tCurrent, float
tEnd) {
return start + (end - start) * (1 + cos(M_PI * tCurrent / tEnd - M_PI)) / 2;
}
```

3.5.2 Integrated Kicks

To move towards faster kick transitions and more natural motion we decided to integrate the kick engine into the walk engine this year. With this we now have have the flexibility to implement a variety of kicks. For now only two types of kicks have been investigated.

Slow Kicks These kicks are modeled off the 2010 kick engine. The idea is that we use coronal rock and a slow step period to transfer our weight onto our support foot. We then use various motions on the non support foot to kick the ball. Currently a forward and side kick have been developed however no work has been done to determine the power of these kicks and as such the kicks are only able to kick with 1 power setting. Future work will need to address this.

For moving the kick leg during these kicks the same function is used for all movements. This function is used to provide smooth interpolation between various joint configurations in the kick. Its salient feature is that is provides a slow acceleration and slow deceleration at the beginning and end of its path to its goal position. This is desirable as we remove any sudden acceleration and thus avoid hiccups in the motors.

Fast Kicks/Dribble Kicks These are kicks that aim to execute within the regular walk cycle. i.e. they are small kicks that execute very quickly. They will probably be used for dribbling. I have only experimented with one type of Fast kick. The idea behind this kick was to place one leg forward and the other leg backwards by approximately 25mm in either direction. With the legs

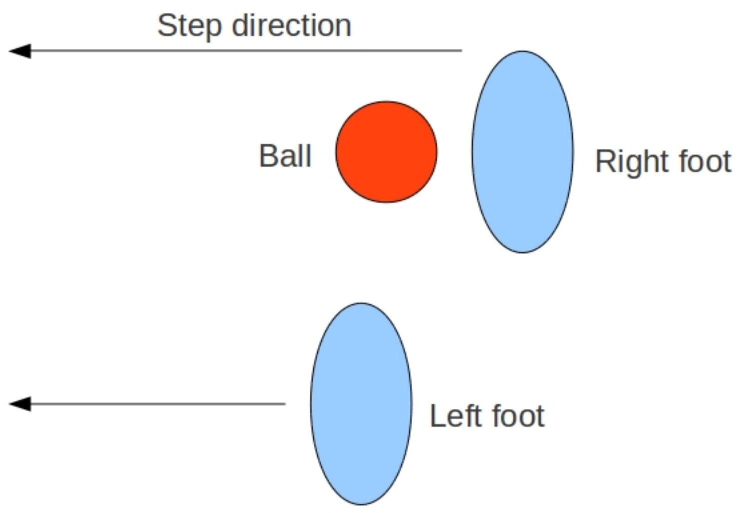

Figure 2: The dribble kick while walking to the left.

in this position one can then perform normal walk movements and dribble the ball either to the left or right. Picture. The problem found when doing this was that the robot became quite unstable. If this instability can be addressed this kick will have potential.

3.5.3 Minor Changes

Phase Switching In the 2010 walk the phase of the walk would switch between support legs even if the lift leg had not been placed on the ground yet. The 2010 walk would switch phases however if the lift leg was placed early.

The change made to this was to only switch phases when the lift leg had been placed on the ground. This is nice as if we get off balance we stop trying to walk and wait for that lift leg to be placed on the ground and thus are less likely to get into unbalanced states. The disadvantage is that we are now much more reliant on foot sensors. If for any reason the foot sensors stop working the walk is now more unreliable. Flashing the touch board is usually necessary in these cases. Another disadvantage of waiting for the foot to be placed before changing phases is that if we rub against another robot in competition and we are stuck with our lift leg on another robot we will stop walking as we do not know when to switch phases. Previously we would start moving our lift leg anyway and thus would squirm free of the other robot. This dependence on the foot sensors may be undesirable.

Goalie Dive The distributed generator[2] has now been altered to add in a special case for the goalie dive. The special case is that we may only transition to a goalie dive from a goalie sit state. And we may only transition out of a goalie dive through the get up routines. As is obvious, this is to ensure we only transition in and out of the goalie dive from the correct states. The actual dive routines used have been adapted from the B-Human dive and further detail on this can be seen in Belinda Teh's Taste of Research report.

3.6 Action Generator

The action generator is a special generator that reads a ".pos" file and generates joint angles from it. Two changes have occurred to the action generator.

No Looping Previous the action generator would keep looping over a pos file. Thus routines would keep repeating. This was not desirable for some actions so this was removed and now the action generator only loops over a pos file once. The action generator must now be reset to loop.

Stiffness To implement the dive we are now able to set the stiffness of individual joints in the pos file. To do this one appends a line starting with a "$"

Algorithm 2 Example pos file that makes left arm go limp while the robot squats into a sitting position.

```
# HY HP LSP LSR LEY LER LHYP LHR LHP LKP LAP LAR RHR RHP
RKP RAP RAR RSP RSR REY RER DUR
0 0 0 30 0 0 -7.5 -5 -60 125 -70 5 5 -60 125 -70 -5 0 -30 0 0 2000
$ 1 1 -1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

after the set of joints one wishes to set the stiffness for. The stiffness desired will effect the stiffness of that joint as it interpolates that particular goal. This was necessary to implement a limp arm such that we may fall on our arm during a goalie dive.

3.7 Plans developed during Taste of Research for Thesis

During Taste of Research plans were also made for the Body Model and Step Planner. Part of this has been implemented inside the Body Model already. For this reason I will now review these plans.

3.7.1 Body Model

The Model The body model aims to use data collected from sensors and the walk engine to estimate a model. The model we have chosen to use is a variation on the inverted pendulum model. Currently the plan is to model an inverted pendulum in 2D for the sagittal plane.

The Centre of Mass of the robot is positioned at the top of the robots hips. This forms the top of the pendulum. The base of the pendulum is to be the centre of the support foot or the centre of pressure on the support foot. We aim to investigate using both the support foot and the centre of pressure as our base as we are unsure as to which will best represent the dynamics of the robot. Using the centre of pressure is more likely to capture the fact that the robots foot is not always flat on the ground and will move the base of the pendulum to account for this. This complicates the model however and is sensitive to the accuracy of the foot sensors. The alternative is to just assume the robots feet are always flat on the ground and thus to use the centre of the feet as the base of the pendulum. This is the approach used by B-Human in their 2011 paper on their walk.

The model will also track the part of the walk cycle that we are currently in. We will track this information so that if we wish to write a predictive A* planner that must iteratively step forward in the walk we will have the information we require.

We will also be tracking a variable called psi in this model. Psi in our model is the angle of the pendulum that has its base at the robot relative coordinate system origin and its top at the centre of mass of the robot. The purpose of psi is to represent the true lean of the robot. For balancing and step planning this variable will be used as a metric to determine how balanced the robot is in any given state. A psi of 0 is a robot that is leaning upright. Note, if we wish to move forward we may also wish to lean the body forward and thus an optimal psi is not always 0.

With this model in mind the challenge now is to estimate it.

Model Estimation To estimate the pendulum we currently aim to use three separate inputs. Firstly, the joint angles are used to determine a forward kinematic solution. This is accurate when the support foot is flat on the ground. This kinematic information will then be combined with data from the accelerometers using a simple kalmen filter. A basic version of this has already been implemented within the BodyModel class.

A planned improvement to this is to only use the kinematic information when we know the robots feet are flat on the ground. Foot sensors may be used to determine when this is the case. Bernhard has also suggested that we may be able to use the change in ZMP (zero moment point) to determine the angular velocity of the robot. This will also be investigated.

3.7.2 Step Planner

This module will use the body model to plan and balance the walk. This will form the bulk of my thesis. I am currently looking at two options.

A* Planner (Model Predictive Control) This will involve adding a predictive function/process update to the body model. This will be used to estimate the position of the model at successive steps. A function will also be added to allow feedback into the bodyModel so that we are able to balance the walk via altering features such as step size and body tilt.

With this in place we are then able to use A* to step forward in an exploratory way. An admissible heuristic such as minimizing the body lean will be used to vector the A* search towards our goal so as to avoid exploring too many states. We will then take a path that leads us to a desirable state, e.g. body lean and angular velocity set to approximately 0.

A nice aspect of the A* Planner is that we can easily expand this to include more complex goals such as allowing the centre of mass to move forward if we are trying to accelerate or turn. It also easily allows for additional methods of feedback such as adding body tilt.

Next step closed form solution This is a method that uses the state of the pendulum to determine the next place the robot should place its foot to maintain the required dynamics and be stable. This has the advantage that the algorithm may be run in constant time as we do not require the iterative method present in A*. The main disadvantage is that we only plan one step ahead. This method is documented in [1].

4 Results

The parameters found for the new parameterized walk show promise. While no quantitative results have been produced it is clear from day to day use that much less stress is present on the robot. The primary indicator to this is that we no long have problems with overheating robots after running them all day.

In its current form, the new parameterized walk is slightly slower as its frequency is set to 0.5 seconds. The old walk cycle operated at 0.38 seconds and for this reason was capable of walking faster with the same step size. To walk at a similar speed work will have to be done to either allow the current walk to interpolate to a faster walk frequency when we wish to walk faster or take larger steps.

The new kicks developed are not yet capable of variable kick power. They also operate at a slow speed when transitioning in and out as time has not yet been spent tuning this. At some point in the future time will have to be spent on these two aspects of the kick engine and we may also build in new stabilization methods in the coronal plane so that we are able to transition more quickly.

More comprehensive results and discussion will be present in my thesis which will be published after robocup 2011.

5 Conclusion

This report has detailed the alterations made to locomotion within 2010-2011 Taste of Research. While the engine looks promising, there is still much work to be done, especially within the kick engine and step planner.

References

[1] Shuuji Kajita, Fumio Kanehiro, Kenji Kaneko, Kiyoshi Fujiwara, Kazuhito Yokoi, and Hirohisa Hirukawa. A realtime pattern generator for biped walking. In *ICRA*, pages 31–37. IEEE, 2002.

[2] Adrian Ratter, Bernhard Hengst, Brad Hall, Brock White, Benjamin Vance, David Claridge, Hung Nguyen, Jayen Ashar, Stuart Robinson, and Yanjin Zhu. runswift team report 2010. 2010.