Black: original comment

Blue: My comments

Red: Original line

Green: replaced line

Yellow: Aditya’s comments

- Related work: Path planning then mapping footsteps onto it decouples the task yes, but is also less optimal, less flexible and under certain circumstances (e.g. not enough safety margin is taken) may not be feasible

No changes. Suggest if you think possible.

- Related work: What does a "wrong solution" look like? What does "ignoring" an invalid case look like in practice? Somehow the robot has to keep walking and you need to command something. What if there is a systematic error and the "invalid" cases don't just resolve themself? What does the robot do?

The system can accept wrong solution once in a few frames as long as it is fast.

The system can accept a less optimal solution once in a few frames as long as it is fast.

As of this comment I will probably address the question of invalid case later in the paper.

- In the related work you mention a number of other works, but don't talk about them at all to in any way compare them to your approach or contrast it somehow (except for Bennewitz and Burgard for example). Mentioning related works isn't very useful unless you say something about them too.

- Related work: The D\* sentence seems incomplete

Added a full stop.

- Related work: Mention a name, not just "[8]"

A comprehensive solution to motion planning is discussed [8]

A comprehensive solution to tiered motion planning specifically for bipedal robots is discussed by Joel Chestnut[8]

- Algorithm 1: What is the effect of "else continue"? There is no loop, so is this supposed to be "return"? In that case, why do you need any "else" at all because it happens anyway.

Removed else continue

- IV-C: Turning when starting walking is said to be solvable by placing a phantom circle, but you don't mention how to place it and/or provide an example.

TO DO: will need some serious work

- Is the camera frame rate 74fps or 50fps?

Such cases are ignored and since the frame rate is as high as 50 fps these instances do not matter much and are taken care of in the subsequent frames.

Such cases are ignored and since the algorithm is run with a frame rate as high as 50 fps these instances do not matter much and are taken care of in the subsequent frames.

- Circles and ellipses aren't polygons because they are not composed of straight edges. I would suggest removing references to 'polygons', because you only use circles and ellipses anyway.

Done, expect for two places. Seems it that way.

- Can the vision estimate the size of an obstacle to avoid the constant radius assumption?

Where in the paper are we mentioning that the radius is constant.

I guess he is saying if the radii are variable then you need to have some basis for choosing a particular radius, so is the basis of that the vision system.

- Equations (1)-(5) are supposed to demonstrate a motion, but seem to be missing a 't' variable. All other parameters in the equations are constants! It is also not explained what Tc is.

@Aditya

The t variable is replaced with a concept because we are only interested in the end-of-step state, not the COM state during a step. Wrote this below equation 1 in my section.

- What happens if the robot is initially standing nice and stable at xi = vi = 0. According to the equations the position and velocity would be constrained to remain zero forever, otherwise it would violate the velocity ratio constraint. How do you deal with this?

@Aditya

Column 2 page 5 start, the point is that these constraints allow me to restrain the motion. I can always use the standard LIPM to handle cases like back walk and starting walk, which I do. These constraints come under our specific implementation.

- The concept of backwards walking could probably be more cleanly solved/explained by being careful with the maths and using +- where necessary where it arises by sqrt.

@Aditya

It can also be handled by standard LIPM, as we have done, as is mentioned in column 2, page 5 start.

- Fig 4: You mention a "stable transition of velocity between straight and curved segments. I cannot see this in Fig 4, where velocity transitions appear to be quite instantaneous.

@Aditya

The transition is stable in the sense that there is no sudden jump from max velocity to circular velocity. Instead, the velocity is reduced progressively as opposed to reducing it in one sudden step. Below equation 10, I have explained all this in a large paragraph. Basically you take one step at a reduced velocity, then reduce it further, until you reach the desired velocity in a number of steps. You do not do it in a single step; in this sense the acceleration/deceleration is smooth, and this is where the constraints come into play, they limit how much we can reduce/increase the velocity in one go.

- Fig 4: Surely the path going underneath the left circle would be the more optimal one?

@Aditya

Not if you are targeting a horizontal orientation. Going underneath the left circle would result in wrong orientation or having to go around the circle. This is evident, so I don’t know how to point it out.

- Fig 4: The left/right leg step sizes seem to be heavily asymmetrical on the straight line segment connecting the two circles. How can this be? What causes this? This can't be particularly desirable.

@Aditya

I just can’t see this part. Please tell me if you can see some asymmetry.

- Fig 6: Tests with >=10 obstacles are quite irrelevant as this is well outside normal operating conditions. What would be interesting is a graph of performance for 0, 1, 2, ..., 10 obstacles, with mean and standard deviations marked (3 waveforms?), so the mean and variability can be seen in the execution time.

WTF! can we make some stupid graph and show some convincing dummy figures?

Man, people like Gautam would come in handy for stuff like this. Personally, I’m not sure how to fake it effectively because I don’t know how a genuine one would look. Would recommend NOT faking it.

- On the side of the footstep planning/motion execution it is generally unclear how the path/footstep plan interacts with the gait. On the lowest lowest level, does the gait in use on the robot receive a gait target vector and that's all? Is this vector a velocity, dimensionless, or a footstep location to step on? Does the gait in any way actually follow this command in a quantitative way? What are the assumptions!

@Aditya

Subsection E is devoted to all of this.

- Is the 3D LIPM used to calculate an open-loop trajectory that is then commanded as an inverse kinematics target to the leg? Or is it only used for the velocity increase ratio?

@Aditya

Subsection E is devoted to all of this.

- How the foot steps are generated from the path is only very vaguely stated, and briefly/in words. A little more insight and explanation would possibly clear up some of the other comments I am making.

@Aditya

Subsection E is devoted to all of this.

- Table 1: What is an "operation"? An operation count is only useful if it approximately relates to performance. So this column seems relatively irrelevant because one method with 57 operations was faster than another with 247 operations, so you can't conclude anything anyway.

Can we take this lite? I copied this data from somewhere!

@Aditya: If you think we can remove do so.

I’m not sure. Depends on how often you refer to it I guess. My advice is to get rid of it. But if you keep it don’t change the data – that would raise suspicion.

- Figures should be at the top of the page, or if this absolutely doesn't work, the bottom, but not inline ('here').

Woah! This guy knows his latex :D

TO DO: rearrange figures

- Values with units can be typeset reliably using siunitx

TO DO: He seriously knows his latex!

- Paragraph indentation (first line of a paragraph) is implemented inconsistently (but should be there)

I fixed as many as I noticed

- "indigenously" is probably not the word you are looking for (twice)

“IT IS”

No, I’m afraid not. Leave it out for now.

- There are regular typographical errors, so a spellchecker would be recommended.

@Aditya: I take it that you have already done this?

To the best of my ability I spellchecked it, will look at the whole thing again.

Physical robot has sensory and localization error, which can make the current perceived position and the target position jump over time. In the worst case, this can make the planner oscillate between a number of possible trajectories and make it much longer for the robot to reach the target position. How can the suggested planner handle such issues?

We still get oscillating paths, but we take history into account esp. in localization where the decay rate is calculated. Need to add this to the paper some place

What we do these days is add certain thresholds to decide whether a new path should be followed by walk, by using a PathUpdateFlag.

-How can the suggested planner be utilized to handle actual RoboCup situations? How big radius around the obstacles should be used, for example for obstacle avoidance challenges? What about the competition situation when the obstacles can move dynamically?

Useless comment  
  
-The walk velocity is limited by the maximum acceleration constraint. Can the author provide a quick test using a physical humanoid robot to support this?

It is not a physical constraint, but a constraint we have introduced by reformulating the LIPM, have made this very explicit in the paper now.