DSU3D Team Description
(3D Soccer Simulation)

Ph. D Shahram Jafari [1], Pouyan Nasrolahi [1], Mehdi Jamaseb [2], Ali Amanzadegan [2], Mehran Khazaeizade [2]

1 Shiraz University
2 Shiraz Bahonar Colleges

pooyan.shz@gmail.com

Abstract
DSU3D is a 3d soccer simulation team. This paper shortly describes the main ideas of the team. These ideas contain: DSU3D 2011 agent structure, agent walk ability using kinematic and planning walking pattern, and finally a decision classifier to be used in strategy layer.
Key words: planning walking pattern, decision classifier, kinematics

I. Introduction

In order to achieve robotic science enhancement in all aspects, robocup has chosen football, and established robocup contests. Simulation league is one of the different robocup leagues that run in a simulated environment. In this way robocup 2D soccer simulation server has been developed. This server caused many of decision controller algorithms and multi agent connection have been implemented and developed. In order to reach robocup long-term goal where humanoid robots can play in physical environment, in 2003 robocup symposium, the 3d soccer simulation introduced as a new field including physics realization. In 2004 robocup, 3d soccer simulation server based on Simspark which is a multi agent simulation system for agent in 3d space, successfully experienced and this league formally established. [1]
Biped motion planning is a challenging problem in humanoid robotics research. There are many efforts in this context and it is still growing. For example we can demonstrate algorithms such as on line and off line walk planning, walking pattern generation, footstep planning, etc.
DSU3D is the Delta3D 2010 team that their members are changed. Other history of team activity describes in team history paper.
The rest of this paper is organized as follows: Section II introduces the agent structure. Section III describes walk algorithms. Section IV presents agent decision structure. And finally we conclude in section V.

II. DSU3D agent structure

In enhancing large programs and especially kind of programs that may have large progression, important problem is to design base structure according to software engineering principles. So after inspecting primary requirements and features that are supposed to be added to the simulation server we started to implement software engineering methods and design main pattern according to object oriented systems that are shortly described as follows:
Some of important metrics in evaluating software quality is correction ability and adaption system. [8] In our provided code, we have tried to consider ability such as system using. Despite of that we have concentrate on simple code structure for easy training after release. Such design gives us this opportunity to considered image processing and synchronization for future use.
DSU3D agent structure is the same as Delta3d 2011 agent. Therefore we will concentrate on 5 major layers and on changed portions.

1. Connection layer: Sends and receives data in network environment between transport layer and the server (Simspark)
2. Data transition layer: Perform data analyzing, saving and conversion between skill and behavior layer into S_Exp.
3. Skill and behavior layer: Consists of robot main abilities such as kick, walk, stand up and their controllers.
4. Rule layer: Determines an agent's duty in a multi agent environment.
5. Decision and Strategy layer: Team's aim and strategy will be minded by this layer.

Performance optimization on these layers is one of the most important applied changes, which has had significant influence on total performance by repairing logical bugs.
Another one of the most important changes, applied on structure was an entity named 'Time Line', which has made some significant changes on 'Skill and Behavior', 'Rule' and 'Strategy and Decision' layers.

Time Line

while the major aim of such competitions is to approach robots behavior to human behavior in fields such as balance minding, movement, making correct decision and etc. therefore robots most be able to predict the around environments status (such as other players and ball), their major activities, etc. this ability enables robots to have significant improvements on their behavior, therefore not only short term targets, but even long-term activities such as can be handled as well.

Time Line is a system that is designed to reach this goal. This system will predict situation of all dynamic objects in the field, will discover future events that will make agent able to reach long-term aims according to environments reactions such as other players movements, ball and etc.

another Time Line usage is to predict actions according to environments feedback that cause better action execution dependent on current situation to reach to the goal.

III. Agent walking

In recent biped motion research, adaptability, on-line, university and intelligence are addressed. Focus of many studies has been walking pattern synthesis. Our current walk is based on planning walking pattern. We are attempting to use new walk method to improve walking operation on NAO robot.

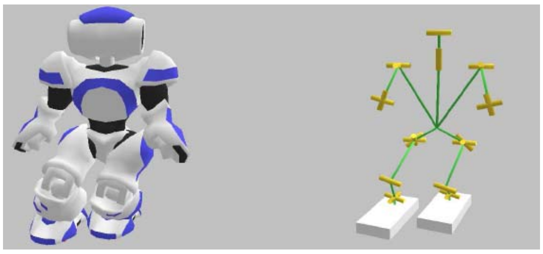


Fig1. Nao Robot

Planning walking pattern

Biped walking is a complex nonlinear dynamic process. It can be considered as a periodic phenomenon. A whole walking cycle consists of two phases: the double-support phase and the single-support phase. Although the double support phase is only about 20% of the duration of a walk cycle, it is important for making the walk more stable and reducing the impact between the swing foot and the ground. However, the single-support phase is the key part of biped walking. During the single-support phase, the robot must place its swing foot in position for the next step while keeping its balance.

We first calculate foot and hip trajectory by a series of offline computation. If both foot trajectories and the hip trajectory are known, all joint trajectories of the biped robot will be determined by kinematic constraints. [2]

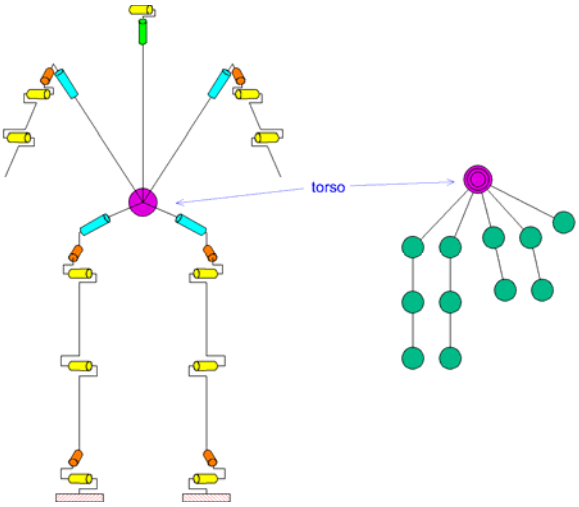


Fig 2. Joints and tree view of robot

To implement these methods we use Apollo 2008 base code and with alter to this pattern we obtain a walk that 1.251 times faster. According to tentative result of experiment on walk we deduced that we can multiply a constant coefficient to value of X-axis of trajectories, but another constraint must be applied, For example increasing acceleration of each joints and decrease agent altitude.

Result of this work can represent as follows:

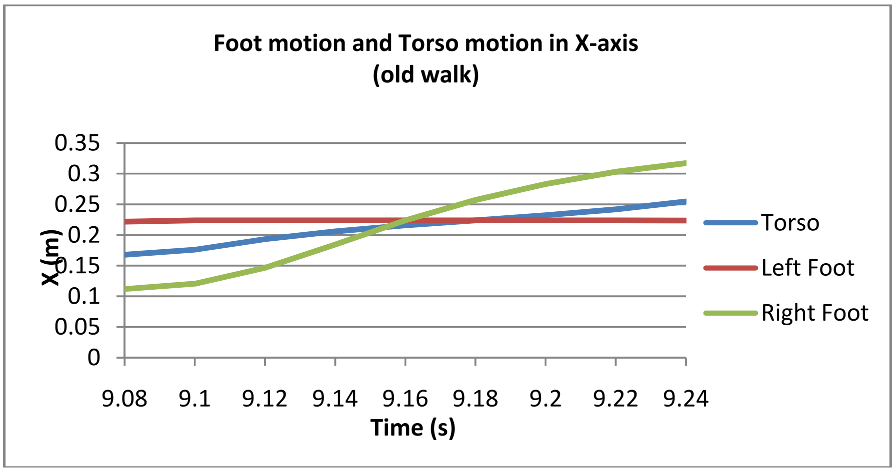


Fig3. Foot motion and torso motion in X-axis (old walk)

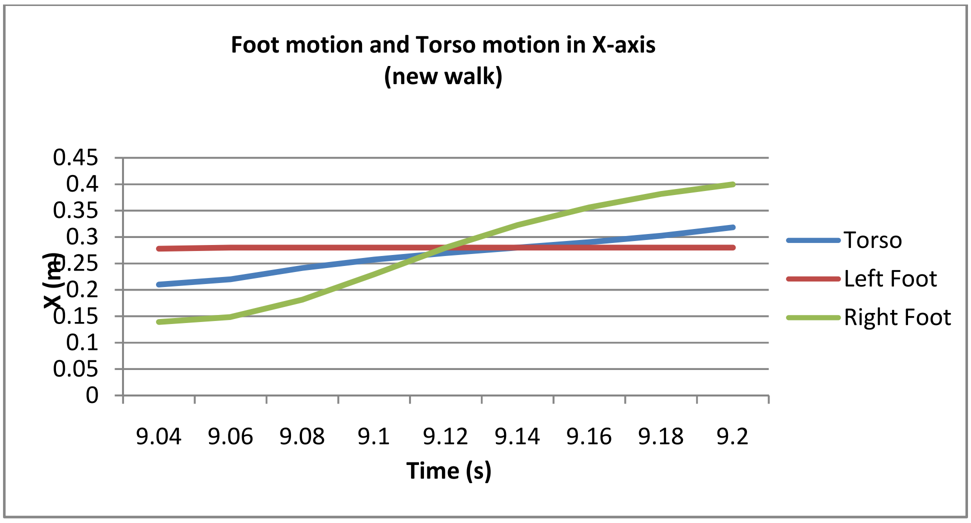


Fig4. Foot motion and torso motion in X-axis (new walk)

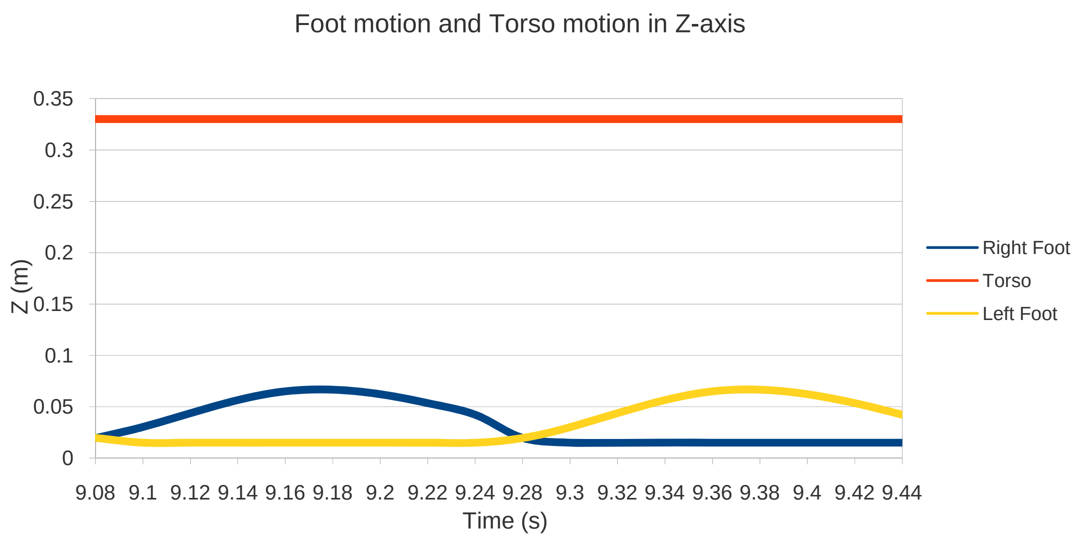


Fig5. Foot motion and torso motion in Z-axis

Omni directional walk

Footstep planner generate robot footstep. To able robot to follow footsteps 3DLIPM model and CoP concept are used. This algorithm simplifies biped walking by considering it as a series of simple support phases. The double-support phase can be planned according to the CoP trajectory. The trajectory of CoP is directly deduced from the foot planner. The CoP is defined to be in the middle of the support area of the support foot during the simple support phase. Using CoP trajectory and LIPM constraint the CoM trajectory can compute. The foot trajectory is also directly deduced from the foot planner. The Cartesian position of the foot is described using homogeneous matrix where the maximum height (0.015 m) is reached in the middle of the simple support phase. The trajectory is computed via a SE3 Interpolation, a cubic spline generalization, which guarantees a smooth trajectory with respect to linear and rotational velocities.

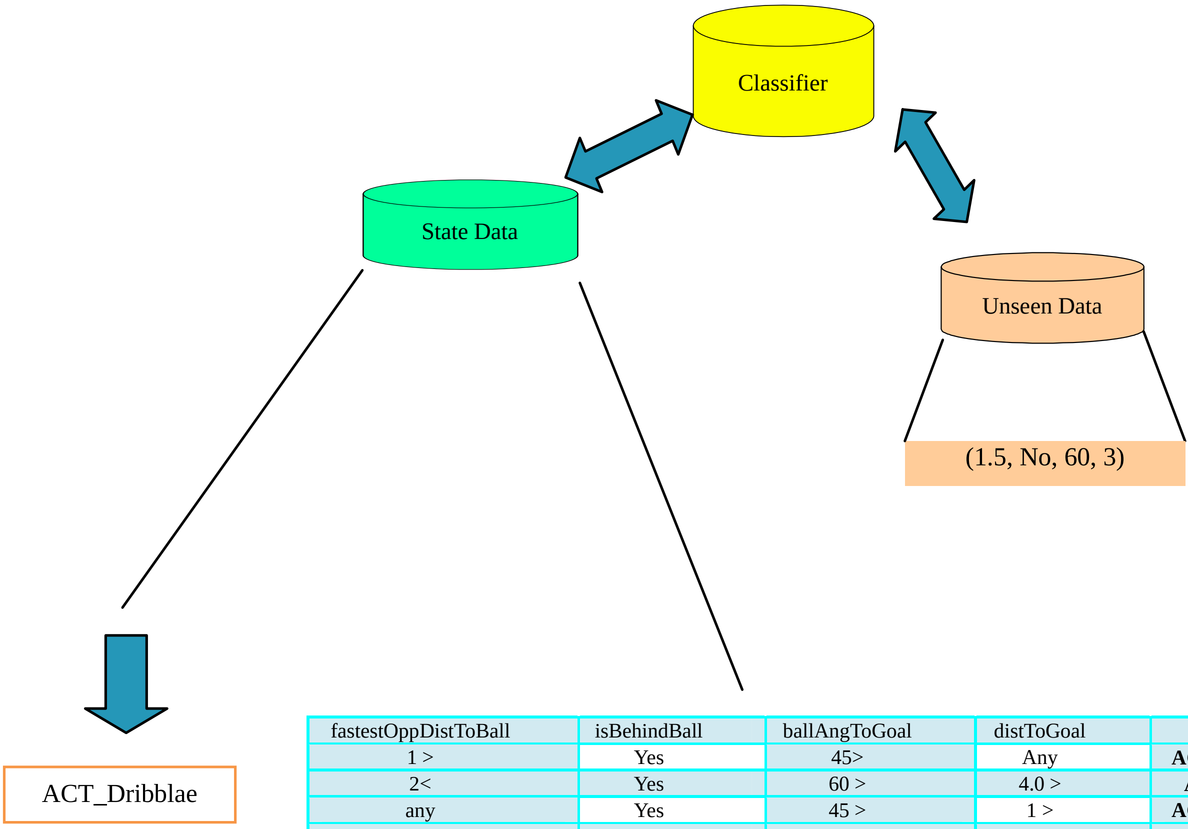
As describe is last section, if both foot trajectories and trunk trajectory are known, all joint trajectories of the biped robot will be determined by kinematics constraints. Football field is a wholly dynamic environment and has millions of states. If we want to have precise decide and cover all the states we must write a condition statement for each state. Herein when we want to add or change a command, total of decision are disassembled, and the commands must be reorganized from top to down. Unfortunately this is not possible. To solve this problem we design a classifier with a series of train state. State data is different state that each player encounter in field while playing, inserted in this model and finally one action such as walk, kick, dribble etc resulted as output.

IV. Decision Classifier

Football field is a wholly dynamic environment and has millions of different states. If we want to have precise decide and cover all the state we must write a condition statement for each state. Herein when we want to add or change a command, total of decisions are disassembled, and the command most be reorganized from top to down. Unfortunately this is not possible. To solve this problem we design a classifier with a series of train state. State data is different state that each player encounter in field, inserted in this model and finally one action such as kick, dribble or walk resulted as output.

V. Conclusion

With implementing these methods, we centralized on improvements of walk stability. We consider this in three aspect, robot control its stability while walking, robot maintain its stability while doing an action and robot do a reaction to remind in stability, to prevent falling down.



fastestOppDistToBall	isBehindBall	ballAngToGoal	distToGoal	**Action**
1 >	Yes	45>	Any	**ACT_Dribble**
2<	Yes	60 >	4.0 >	**ACT_Kick**
any	Yes	45 >	1 >	**ACT_Dribble**
2<	Yes	60<	Any	**ACT_Pass**
2<	Yes	Any	9<	**ACT_Kick**

VI. References

[1] Robocup3D Server Develop Group. RoboCup Soccer Server 3D Manual. 2007.

[2] Qiang Huang, Kazuhito Yokoi, Shuuji Kajita, Kenji Kaneko, Hirohiko Arai, Noriho Koyachi, Kazuo Tanie, “Planning Walking Patterns for a Biped Robot”, IEEE TRANSACTIONS ON ROBOTICS AND AUTOMATION, VOL. 17, NO. 3, JUNE 2001, pp. 280-289

[3] Jinsu Liu, Manuela Veloso, “Online ZMP Sampling Search for Biped Walking Planning”, 2006.

[4] Kajita, S. and Kanehiro, F. et al., The 3D Linear Inverted Pendulum Mode: A simple modeling for a biped walking pattern generation, in *Proce. of 2001 International Conference on Intelligent Robots and Systems*, vol. 1, 2001, pp. 239-246.

[5] Sugihara, T. and Nakamura, Y. et al., Realtime Humanoid Motion Generation through ZMP Manipulation based on Inverted Pendulum Control, in *Proce. of 2002 ICRA*, vol. 2, 2002, pp. 1404-1409

[6] A.Goswami, “Foot-Rotation indicator (FRI) point: Anew Gait Planning Tool to Evaluate Postural Stability of Biped Robots” Proc. IEEE ICRA, Cetroit, 1999

[7] Kurt .O, Erbayur .K, “Biped Robot Reference Generation with Natural ZMP Trajectory”, The 9th IEEE International Workshop on Advance motion Control, 2005. ”

[8] M. Vukobratovic et al, “Biped Localization Dynamic, Stability, Control and Application”, springer-verlag 1990.

[9] Pressman, Ragers , “Software Engineering ”, a practitioner’s approach ,5th , ed ,2001.

[10] Tajima,D and T . Matsubra, “The computer software industry in Japan”, Computer , May 1981, p.96.

[11] Shineo Shingo, Zero Quality Control; Source inspection and the poko-yoke system, productivity ,1986.