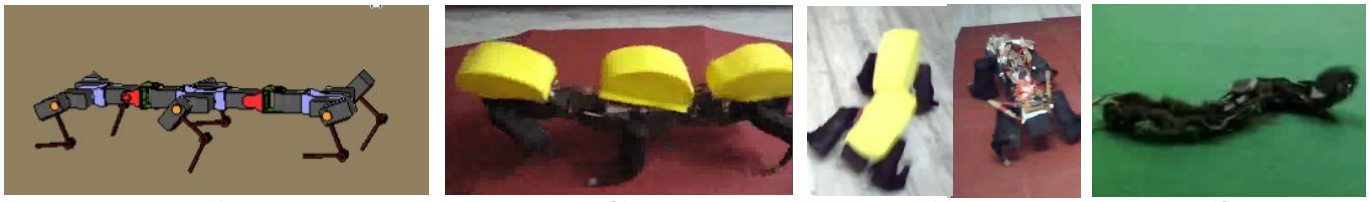
“Mod-Leg” A modular Legged robotic system

Sriranjan Rasakatla
Robotics Research Lab
IIIT-H
infibit@gmail.com

K Madhava Krishna
Robotics Research Lab
IIIT-H
mkrishna@iiit.ac.in

Bipin Indurkhya
Robotics Research Lab
IIIT-H
bipin@iiit.ac.in



(a) (b) (c) (d)

Figure 1: (a) Shows the generic modular legs with 2 d.o.f spinal joints linkage. (b) Hexapod configuration climbing over a slope (c) 4 legged dog configuration walking and climbing over a slope (d) Snake configuration with just the robotic spine and no legs.

1. Abstract

The Modular Legged robotic system [1] “Mod-Leg” presented here has been bio-inspired from a Snake’s vertebrae and a caterpillar’s legged structure. The system can be configured to a 4-legged robotic dog, a hexapod, a caterpillar and a Snake robot. This robot’s novel design achieves compliance to the terrain using a combination of legs and electronically actuated universal spine. A unique simulator has been designed for this purpose. Some of the things we learned while developing this robotic system have been presented below.

2. System Features.

a) We developed an API (modleg API) for Sketchy Physics. The integrated environment of Google Sketchup with Sketchy Physics had limitations due to the inbuilt minimal ruby interface. This avoided programming the virtual models/characters in the physics simulation for full fledged kinematic algorithms. So a ruby plugin which would dump the physics engine parameters like motor velocities, torque, joints exerted by the piston, servo angle values into a shared memory was written. A DLL was then written with functions to read and write the values from the shared memory. Using this ruby plugin and a DLL as the bridge a game developer or a researcher looking for a good design and physics simulation package could read the values from the shared memory. This approach provided an API to sketchy physics independent of the language (as all one needs to do is read and write into the shared memory). So a programmer using Java, C, C++, Ruby or python etc could use the API. This was formerly not possible with Google Sketchup running sketchy Physics,

b) Also using the design idea of a modular leg with 2 d.o.f spinal joints we developed templates for a Snake robot, a caterpillar robot, a 6 legged Hexpod and a 4 legged robotic dog. This can be adopted in games and simulation environments

c) The virtual model in the simulator was closely modeled to the real prototype in both design and API. This ensured that the kinematic algorithms tested in the simulation engine worked exactly the same on the prototype robot during real time tests. The common API ensured that no further porting was required to transfer the sequence of actions of the virtual model to the real robot. This unified development approach greatly reduced design time, development time and saved resources (battery power and replacements for worn out parts) required for the real time tests. We feel this approach would also help in bringing virtual models/characters to robotic reality (life). Also simulations were done to make the robot swim in water (a buoyant environment).The tripod gait in hexapod configuration of the robot did not show good forward velocity. The integrated simulation and design environment enabled us to observe and change the paddle design then and there it self. Also the swimming algorithm was changed using the modleg API to get more forward velocity.

d) One can interact with the virtual robot and the real prototype from a single point GUI. We call this the universal simulator and controller. An inverse kinematic stabilization algorithm on terrain with changing and unpredictable slopes was written for the robot. The algorithm adjusts the hip to toe lengths of the robot for making it stable. An accelerometer was used as an input device to vary the slope of a test platform on which the robotic dog was resting in the physics simulation. Later during real time tests when the accelerometer was mounted on the platform it was observed that the real robot stabilized itself with negative gains. This was possible because of the universal controller built developed on a common API.

e) The maximum possible stiffness of the spinal cord of the robot in 6 legged and caterpillar configuration was set to a threshold value. This universal spine made the virtual models show compliance to uneven terrain. This did not require complex inverse kinematic programming by taking into consideration the varying geometries of the terrain. The compliance was achieved using a combination of legs and the universal spinal cord. This idea of compliance can thus be used in games for templates like Snakes, caterpillars and hexapods. The compliance to terrain was also observed on the real robot while climbing slopes up and down.

f) The robot can be controlled from a Laptop’s USB port using an Zigbee RF modem. The robot also uses CRC based error checking protocol to overcome communication errors and still move under noisy RF conditions. The robot is equipped with onboard lithium polymer power with over current and over voltage protection for the servos.

3.References :

[1]Demo reel of the robot is here:
http://www.youtube.com/watch?v=Wf9Y-lB_t7s

SIGGRAPH 2010, Los Angeles, California, July 25 – 29, 2010.
ISBN 978-1-4503-0210-4/10/0007