Design and implementation of Stair Climbing Robo controlled by Bluetooth

[1]K.Parvateesam,[2]T.Anjaiah
[1]Assistant Professor,[2]Assistant Professor
[1]Department of ECE,
[1]Aditya College of Engineering and Technology, Kakinada, AP, India.

Abstract : In today's life, technology concerned with robots plays an important role in many fields because they are used to operate in hazardous and urban environments, for security, in traffic system, rescue mission as well as military operations. Some of the e robots are designed to operate only on natural terrains, but it can also use for rough terrains and artificial environments including stairways. This article represents the mechanism of how will robot climb the stairs carrying load. Its mechanical design is suitable with front wheel and back wheel driven by DC motor for climbing stairs. Although many robots had been introduced earlier have some problems like need of special device or software to control the robot etc. This design allows an advance method for robotics control using the mechanical links. Until recent years, the stair climbing robots are designed with vast hardware and robots are equipped with chain roller to climb stairs or to move on a flat surface. The mechanical design of the robot contains the fixed and flexible links of wheel legs instead of chain roller moves relative to each other to generate high friction with stairs. There is a lot of scope for improvement and this mechanism can be further modified and used in various other applications such as carrying heavy loads and thus further reducing human effort. Another scenario where this mechanism can be employed is during disaster management. A camera can be fitted on the robot to have a wide field of view of the affected areas which can further help in search and rescue operations. This robot can further be integrated with mobile devices to process the images fed by the camera and act accordingly to the stairs.

***IndexTerms* – Stair Climbing, Bluetooth.**

I INTRODUCTION

Over the past decade, the stair climbing robot suspension design has become a verified mobility application known for its greater vehicle stability and obstacle-climbing capability. This stair climbing robot is known as Roker Bogie structure. Following numerous technology and research rover implementations, the system was successfully flown as part of Mars Pathfinder’s Sojourner rover. When the Mars Exploration Rover (MER) Project was first proposed, the use of a rocker-bogie suspension was the obvious choice due to its extensive heritage. The challenge posed by MER was to design a lightweight rocker-bogie suspension that would permit the mobility to stow within the limited space available and install into a configuration that they could then safely use to egress from the lander and explore the Martian surface. Simplicity is something you need when you build a robot. There were numerous cases when a suspension system could not be evaded even though in maximum cases you'd never require a suspension system. The term “rocker” describes the rocking aspect of the larger links present each side of the suspension system and balance the bogie as these rockers are connected to each other and the vehicle chassis through a selectively improved differential. And the term “bogie” refers to the links that have a drive wheel at each end. Bogies were commonly used as load wheels in the tracks of army tanks as idlers distributing the load over the terrain. Both applications now select trailing arm suspensions. The design of rocker-bogie has no springs or stub axles for each wheel, permitting the rover to climb over hurdles, such as rocks, that is up to twice the wheel's diameter in size while keeping all six wheels on the ground. Tilt stability is limited by the height of the centre of gravity with any suspension system.

A Rocker

The term “rocker” comes from the rocking aspect of the larger links on each side of the suspension system. These rockers are connected to each other.

B Bogie

The term “bogie” refers to the links that have a drive wheel at each end. It is the cabin which is used for placing the components.

II HARDWARE SPECIFICATIONS

Arduino is used for building different types of electronic circuits easily using of both a physical programmable circuit board usually microcontroller and piece of code running on computer with USB connection between the computer and Arduino.Programming language used in Arduino is just a simplified version of C++ that can easily replace thousands of wires with words.

2.1Basic description

NASA recently started an ambitious exploration program of Mars. Pathfinder is the first over explorer in this program. Future rovers will need to travel several kilometers over periods of months and manipulate rock and soil samples. The term “rocker” describes the rocking aspect

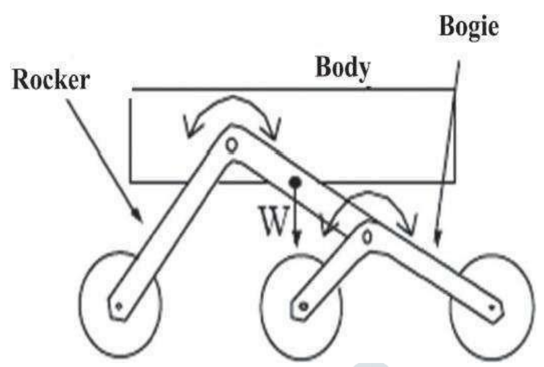


Fig 2.1 Line Diagram of Rocker Bogie Mechanism

the larger links present each side of the suspension system and balance the bogie as these rockers are connected to each other and the vehicle chassis through a modified differential.

As accordance with the motion to maintain center of gravity of entire vehicle, when one rocker moves up- ward, the other goes down. The chassis plays vital role to maintain the average pitch angle of both rockers by allowing both rockers to move as per the situation. The physics of these rovers is quite complex. To design and control these analytical models of how the rover interacts with its environment are essential. Models are also needed for rover action planning. Simple mobility analysis of rocker-bogie vehicles have been developed and used for design evaluation in the available published works. The rocker-bogie configuration is modeled as a planer system. Improving the performances of a simpler four wheel rover has also been explored.

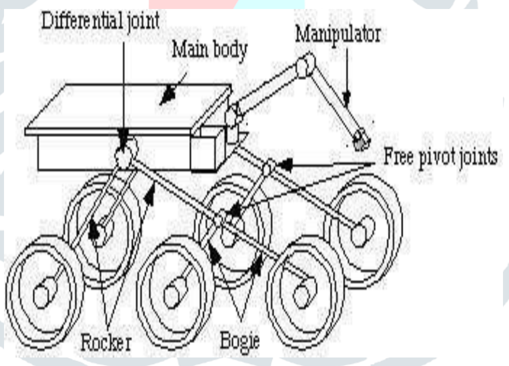


Fig 2.2 Three Dimensional view of Rocker Bogie Mechanism

2.2 HC-05 BLUETOOTH MODULE

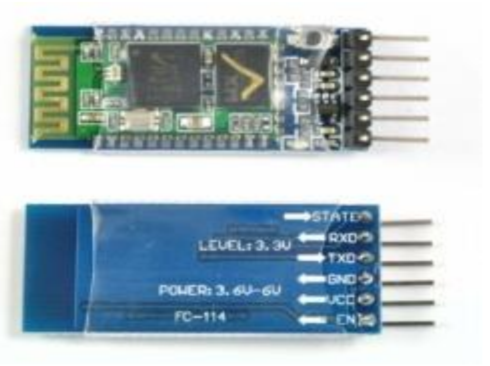


Fig 2.3 HC-05 module

HC-05 module is an easy to use **Bluetooth SPP (Serial Port Protocol) module**, designed for transparent wireless serial connection setup. The HC-05 Bluetooth Module can be used in a Master or Slave configuration, making it a great solution for wireless communication. This serial port bluetooth module is fully qualified **Bluetooth V2.0+EDR (Enhanced Data**

Rate)3Mbps Modulation with complete 2.4GHz radio transceiver and baseband. It uses **CSR Bluecore 04-**External single chip bluetooth system with CMOS technology and with AFH (Adaptive Frequency Hopping Feature).This is a MASTER/SLAVE module. By default the factory setting is SLAVE. The Role of the module (Master or Slave) can be configured only by AT COMMANDS. The slave modules cannot initiate a connection to another Bluetooth device, but can accept connections. Master module can initiate a connection to other devices. The user can use it simply for a serial port replacement to establish connection between MCU and GPS.

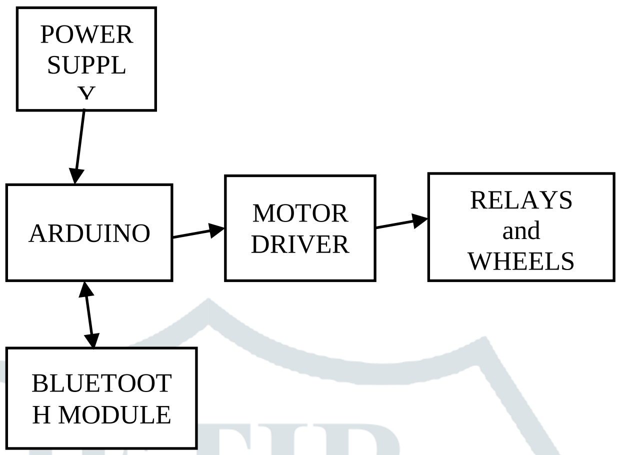


Fig 2.4 : Block Diagram

2.3 Working principle

There is stepwise working procedure for the movement of the robot.4 steps are there for taking the movement of the robot. **A.** Lifting the front Part Initially robot is horizontally placed. When there is stair available in front of the robot it will lift its both arms upward then place those arms on the stair. Then rotate its wheels forward. **B.** Lifting the middle part of the robot while moving forward After moving the front part of the robot it will lift its middle or back part by just controlling the movement of servo motor by the programming. **C.** Lifting the back part of the robot After moving the back part it will start moving wheels forward for making it to move forward. After getting sufficient space on next stair it will then find for that another stair is there or not. If there is another step available then it will repeat the above procedure. Working of the robot takes place step wise. robot comes to rest momentarily after each step. four steps for climbing the stairs are

1. Robot wheel touches the step
2. Lifting the front part.
3. Lifting the back part of the robot.
4. Following the above steps the robot proceeds.
5. It can also be used for descending of steps.
6. Robot wheel touches the step

Initially the robot is in horizontal position and when the robot touches the first step, the upper wheel is ready to move upward to lift the front part of the robot.

2.4 Steps for making the robot

1. First design the steel frame as per the dimensions with adjustable mechanism.
2. structure of the robot should be rigid.
3. five D.C motors are attached to the end of the frame.
4. Four 12v batteries are connected in series for constant movement of wheels.
5. positive connection is upto12v, and negative connection is up to 6v.
6. Positive connection is for forward motion and negative connection is for backward motion.
7. These connections are linked to circuit board for the robot motion.
8. robot can be controlled by remote which is connected to circuit.
9. Remote has a switch which is helpful for the backward and forward motion of robot.

III SOFTWARE IMPLEMENTATION

Integrated development environments are designed to maximize programmer productivity by providing tight-knit components with similar user interfaces. IDEs present a single program in which all development is done. This program typically provides many features for authoring, modifying, compiling, deploying and debugging software. This contrasts with software development using unrelated tools, such as vi, GCC or make. One aim of the IDE is to reduce the configuration necessary to piece together multiple development utilities, instead it provides the same set of capabilities as one cohesive unit. Reducing setup time can increase developer productivity, especially in cases where learning to use the IDE is faster than manually integrating and learning all of the individual tools. Tighter integration of all development tasks has the potential to improve overall productivity beyond just helping with setup tasks. For example, code can be continuously parsed while it is being edited, providing instant feedback when syntax errors are introduced. Allowing developers to debug code much faster and easier with an IDE. Some IDEs are dedicated to a specific programming language, allowing a feature set that most closely matches the programming paradigms of the language. However, there are many multiple-language IDEs. While most modern IDEs are graphical, text-based IDEs such as Turbo Pascal were in popular use before the widespread availability of windowing systems like Microsoft Windows and the X Window System (X11). They commonly use function keys or hotkeys to execute frequently used commands or macros. IDEs initially became possible when developing via a console or terminal. Early systems could not support one, since programs were prepared using flowcharts, entering programs with punched cards (or paper tape, etc.) before submitting them to a compiler. Dartmouth BASIC was the first language to be created with an IDE (and was also the first to be designed for use while sitting in front of a console or terminal). Its IDE (part of the Dartmouth Time Sharing System) was command-based, and therefore did not look much like the menu-driven, graphical IDEs popular after the advent of the Graphical User Interface. However it integrated editing, file management, compilation, debugging and execution in a manner consistent with a modern IDE.

3.1 SYNTAX HIGHLIGHTING

The IDE editor usually provides syntax highlighting, it can show both the structures, the language keywords and the syntax errors with visually distinct colors and font effects

Refactoring

Advanced IDEs provide support for automated refactoring.

Version control

An IDE is expected to provide integrated version control, in order to interact with source repositories

Debugging

IDEs are also used for debugging, using an integrated debugger, with support for setting breakpoints in the editor, visual rendering of steps, etc.

Code search

IDEs may provide advanced support for code search: in order to find class and function declarations, usages, variable and field read/write, etc. IDEs can use different kinds of user interface for code search, for example form-based widgets and natural-language based interfaces.

3.2 Visual programming

Visual programming is a usage scenario in which an IDE is generally required. Visual Basic allows users to create new applications by moving programming, building blocks, or code nodes to create flowcharts or structure diagrams that are then compiled or interpreted. These flowcharts often are based on the Unified Modeling Language. This interface has been popularized with the Lego Mindstorms system, and is being actively pursued by a number of companies wishing to capitalize on the power of custom browsers like those found at Mozilla. KTechlab supports flowcode and is a popular opensource IDE and Simulator for developing software for microcontrollers. Visual programming is also responsible for the power of distributed programming (cf. LabVIEW and EICASLAB software). An early visual programming system, Max, was modeled after analog synthesizer design and has been used to develop real-time music performance software since the 1980s. Another early example was Prograph, a dataflow-based system originally developed for the Macintosh. The graphical programming environment "Grape" is used to program qfix robot kits. This approach is also used in specialist software such as Openlab, where the end users want the flexibility of a full programming language, without the traditional learning curve associated with one.

Language support

Some IDEs support multiple languages, such as GNU Emacs based on C and Emacs Lisp, and IntelliJ IDEA, Eclipse, MyEclipse or NetBeans, all based on Java, or MonoDevelop, based on C#, or PlayCode.Support for alternative languages is often provided by plugins, allowing them to be installed on the same IDE at the same time. For example, Flycheck is a modern on-the-fly syntax checking extension for GNU Emacs 24 with support for 39 languages.

3.3 Attitudes across different computing platforms

Unix programmers can combine command-line POSIX tools into a complete development environment, capable of developing large programs such as the Linux kernel and its environment. In this sense, the entire Unix system functions as an IDE.The free software GNU tools (GNU Compiler Collection (GCC), GNU Debugger (gdb), and GNU make) are available on many platforms, including Windows. The pervasive Unix philosophy of "everything is a text stream" enables developers who favor command-line oriented tools to use editors with support for many of the standard Unix and GNU build tools, building an IDE with programs like Emacs or Vim. Data Display Debugger is intended to be an advanced graphical front-end for many text-based debugger standard tools. Some programmers prefer managing makefiles and their derivatives to the similar code building tools included in a full IDE. For example, most contributors to the PostgreSQL database use make and gdb directly to develop new features. Even when building PostgreSQL for Microsoft Windows using Visual C++, Perl scripts are used as a replacement for make rather than relying on any IDE features. Some Linux IDEs such as Geany attempt to provide a graphical front end to traditional build operations. On the various Microsoft Windows platforms, command-line tools for development are seldom used. Accordingly, there are many commercial and non-commercial products. However, each has a different design commonly creating incompatibilities. Most major compiler vendors for Windows still provide free copies of their command-line tools, including Microsoft (Visual C++, Platform SDK, .NET Framework SDK, nmake utility). IDEs have always been popular on the Apple Macintosh's classic Mac OS and macOS, dating back to Macintosh Programmer's Workshop, Turbo Pascal, THINK Pascal and THINK C environments of the mid-1980s. Currently macOS programmers can choose between native IDEs like Xcode and open-source tools such as Eclipse and Netbeans. ActiveState Komodo is a proprietary multilanguage IDE supported on macOS.

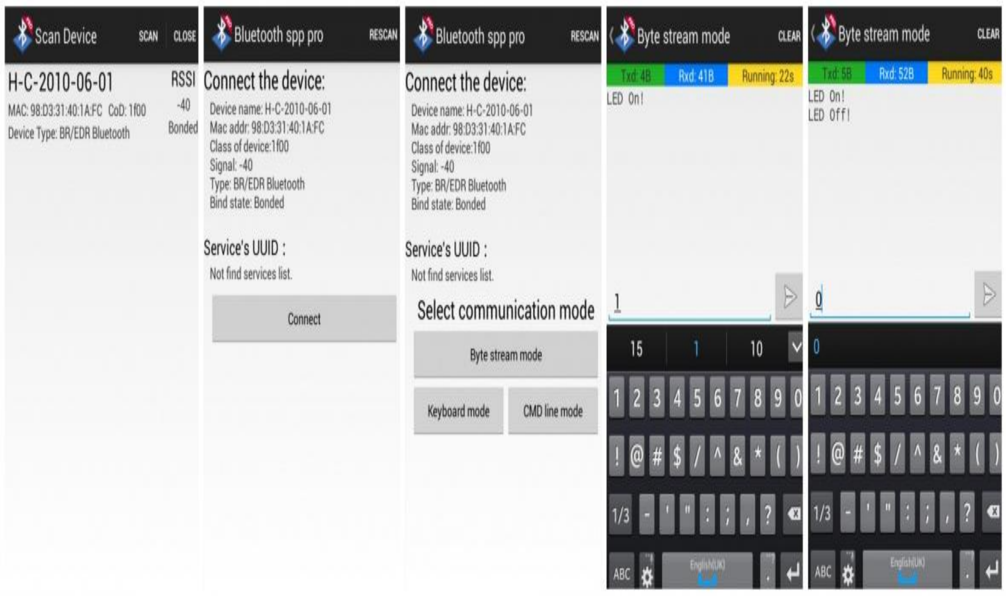
Robot Applications Include Military and Security Robots

This technology has applications as military robots or security robots in urban environments where stair climbing and agile operation is an essential part of the mission. The robot is not only a stair climbing robot but operates in terrain where wheeled robots would operate. Stair Climbing Robot with Innovative Wheel Design, Fast, Portable Stair Climbing Robot, Wheeled and tracked vehicles, while cheaper than bipedal or jumping robots, often have problems with slippage. The Loper robotic platform has solved this problem with an innovative wheel design for mobile robots consisting of three lobes connected by to a central hub that essentially act as cogs for the purpose of stair climbing. The design enables fast stair climbing at a reduced platform weight. The operation is capable of achieving speeds of up to 8km/hr on flat surfaces and stair climbing speed of 4.3 body lengths/second. The chassis design provides a light, flexible and rugged platform that is ideal for agile operation. The platform can be outfitted with image sensors, infrared sensors, or chemical detectors. Loper is also capable of sustained motion for extended periods of operation and is more portable than the Packbot.

Features and benefits of stair climbing robot

- ❖ High stair climbing speed: Capable of achieving speeds of up to 8km/hr and a climbing speed of 4.3 body lengths/second
- ❖ Lightweight and rugged design for increased portability
- ❖ Platform can be outfitted with an array of sensor capabilities
- ❖ More portable than the Packbot.
- ❖ Capable of sustained motion for extended periods of operation.

IV RESULT



Scan Device SCAN CLOSE
H-C-2010-06-01 RSSI -40 Bonded
MAC: 98:D3:31:40:1A:FC CoD: 1f00
Device Type: BR/EDR Bluetooth

Bluetooth spp pro RESCAN
Connect the device:
Device name: H-C-2010-06-01
Mac addr: 98:D3:31:40:1A:FC
Class of device:1f00
Signal: -40
Type: BR/EDR Bluetooth
Bind state: Bonded
Service's UUID :
Not find services list.
Connect

Bluetooth spp pro RESCAN
Connect the device:
Device name: H-C-2010-06-01
Mac addr: 98:D3:31:40:1A:FC
Class of device:1f00
Signal: -40
Type: BR/EDR Bluetooth
Bind state: Bonded
Service's UUID :
Not find services list.
Select communication mode
Byte stream mode
Keyboard mode CMD line mode

Byte stream mode CLEAR
Txd: 4B Rxd: 41B Running: 22s
LED On!
1

Byte stream mode CLEAR
Txd: 5B Rxd: 52B Running: 40s
LED On!
LED Off!
0

Figure 3.15 Bluetooth application

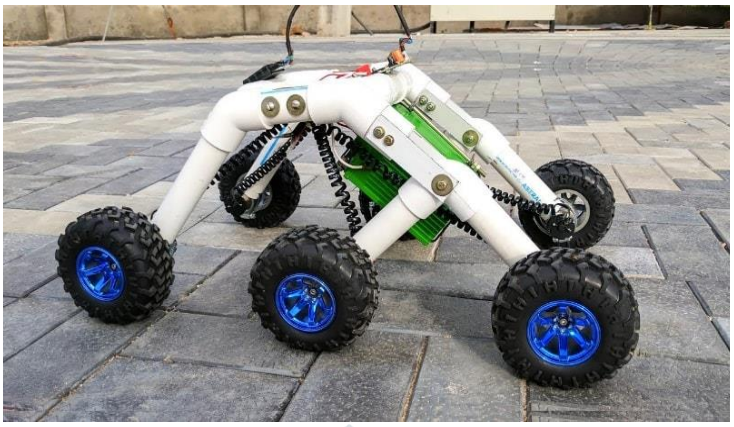


Figure 5.2: Robot is in initial position

Initially the robot is in horizontal position and when the robot touches the first step, the upper wheel is ready to move upward to lift the front part of the robot.

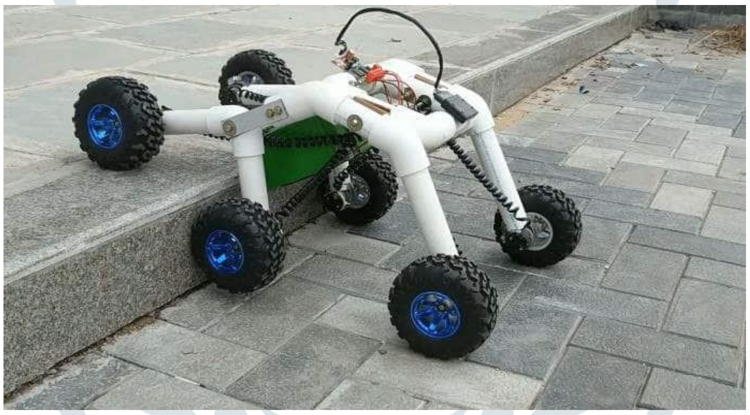


Figure 5.3: Lifting the front part

Lifting the front part

By switching on, the motor starts rotating the upper wheel which lifts the front part of the robot to a certain height. front wheels will move forward through the step and also helps the robot to climb the stair.

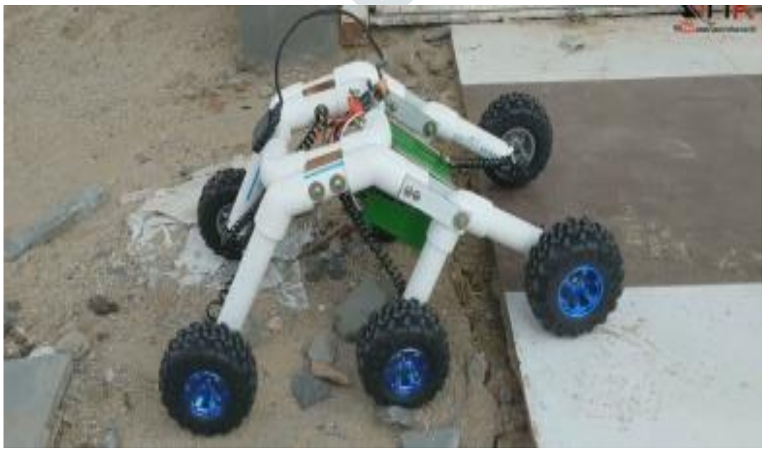


Figure 5.3: Lifting the Back part

Lifting the back part of the robot:

After lifting the front part of the robot, the rear wheel touches the step and moves forward. After climbing all the steps, it will reaches to its initial position.

V CONCLUSION

This work shows how rover bogie system works on different surfaces. As per the different weight acting on link determines torque applied on it. By assuming accurate stair dimensions, accurately dimensioned rover bogie can climb the stair with great stability. The design and manufactured model CAN climb the angle up to 90˚. During stair climbing test for length less than 375 mm (15inch) system cannot climb the stair. It can be possible to develop new models of rocker bogie which can climb the stairs having low lengths.

VI ACKNOWLEDGEMENT

We express our sincere gratitude and heartfelt thanks to the under stated persons for the successful completion of our project on HAND TALK USING FOUR FLEX SENSORS WITH VOICE OUTPUT. We feel much grateful to Mr.D.KISHORE, M.Tech,(Ph.D.), Associate Professor and Head of the department of ECE, who provided vital information and helped in completion of our work successfully. We express our sincere thanks to Dr.T.RAMA KRISHNA RAO, M.Tech,Ph.D., Principal of Aditya College of Engineering & Technology, for his enriching thoughts and profound knowledge, which brought our work to its completion. We own our sincere gratitude to management of ADITYA COLLEGE OF ENGINEERING & TECHNOLOGY for their help directly and indirectly in completion of this work.

REFERENCE

[1] M. Eich, F. Grimminger, and F. Kirchner, "Adaptive stair–climbing behaviour with a hybrid legged–wheeled robot," in 11th International Conference on Climbing and Walking Robots and the Support Technologies for Mobile Machines (CLAWAR), Coimbra, Portugal, September 8–10 2008, to appear.
[2] M. Eich, F. Grimminger, S. Bosse, D. Spenneberg, and F. Kirchner, "Asguard: A hybrid legged wheel security and sar–robot using bio– inspired locomotion for rough terrain," in IARP/EURON Workshop on Robotics for Risky Interventions and Environmental Surveillance, Benicssim, Spain, January 7–8 2008.
[3] U. Saranli, M. Buehler, and D. Koditschek, "Rhex: A simple and highly mobile hexapod robot," The International Journal of Robotics Research, vol. 20, no. 7, pp. 616–631, July 2001.
[4] J. Quinn, R.D.and Offi, D. Kingsley, and R. Ritzmann, "Improved mobility through abstracted biological principles," in IEEE/RSJ International Conference on Intelligent Robots and System, Volume 3, 2002, pp. 2652– 2657.
[5] E. Moore and M. Buehler, "Stable stair climbing in a simple hexapod robot," 2001. [Online]. Available: "citeseer.ist.psu.edu/moore01stable.html
[6] H. Komsuoglu, D. McMordie, U. Saranli, D. Moore, M. Buehler, and D. E. Koditschek, "Proprioception based behavioral advances in hexapod robot," in Proceedings of the IEEE Int. Conf. on Robotics and Automation, Seoul, Korea, May 21-26, 2001, pp. pp 3650 – 3655.
[7] G. C. Haynes and A. Rizzi, "Gaits and gait transitions for legged robots," in Proceedings of the 2006 IEEE International Conference on Robotics and Automation (ICRA '06), May 2006.
[8] T. Allen, R. Quinn, R. Bachmann, and R. Ritzmann, "Abstracted biological principles applied with reduced actuation improve mobility of legged vehicles," in Proceedings. 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS2003), Volume 2, 2003, pp. 1370– 1375.
[9] J. Ayers, "A conservative biomimetic control architecture for autonomous underwater robots," in Neurotechnology for Biomimetic Robots, Ayers, Davis, and Rudolph, Eds. MIT Press, 2002, pp. 241– 260. [10] F. Kirchner, D. Spenneberg, and R. Linnemann, "A biologically inspired approach towards robust real world locomotion in an 8-legged robot," in Neurotechnology for Biomimetic Robots, J. Ayers, J. Davis, and A. Rudolph, Eds. Cambridge, MA, USA: MIT-Press, 2002.
[11] A. Ijspeert and J. Kodjabachian, "Evolution and development of a central pattern generator for the swimming of a lamprey," 1998. [Online]. Available: citeseer.ist.psu.edu/article/ijspeert98evolution.html [12] A. H. C. Y. Fukuoka, H. Kimura, "Adaptive dynamic walking of a quadruped robot on irregular terrain based on biological concepts," International Journal of Robotics Research, vol. 22, pp. 187–202, 2003. [13] D. Spenneberg and F. Kirchner, The Bio-Inspired Scorpion Robot: Design, Control & Lessons Learned. Houxiang Zhang, I-Tech Education and Publishing, 2008, pp. 197–21