5-2007

An Emotional Mimicking Humanoid Biped Robot and its Quantum Control Based on the Constraint Satisfaction Model

Quay Williams
Portland State University

Scott Bogner
Portland State University

Michael Kelley
Portland State University

Carolina Castillo
Portland State University

Martin Lukac
Portland State University

See next page for additional authors

Citation Details

Williams Q., Bogner S., Kelley M., Castillo C., Lukac M., Kim D. H., Allen J., Sunardi M., Hossain S., Perkowski M. "An Emotional Mimicking Humanoid Biped Robot and its Quantum Control Based on the Constraint Satisfaction Model," 16th International Workshop on Post-Binary ULSI Systems, 2007

Authors

Quay Williams, Scott Bogner, Michael Kelley, Carolina Castillo, Martin Lukac, Dong Hwa Kim, Jeff S. Allen, Mathias I. Sunardi, Sazzad Hossain, and Marek Perkowski

AN EMOTIONAL MIMICKING HUMANOID BIPED ROBOT AND ITS QUANTUM CONTROL BASED ON THE CONSTRAINT SATISFACTION MODEL

Intelligent Robotics Laboratory, Portland State University
Portland, Oregon.

Quay Williams, Scott Bogner, Michael Kelley, Carolina Castillo, Martin Lukac, Dong Hwa Kim, Jeff Allen, Mathias Sunardi, Sazzad Hossain, and Marek Perkowski

Abstract

The paper presents a humanoid robot that responds to human gestures seen by a camera. The behavior of the robot can be completely deterministic as specified by a Finite State Machine that maps the sensor signals to the effector signals. This model is further extended to the constraints-satisfaction based model that links robots vision, motion, emotional behavior and planning. One way of implementing this model is to use adiabatic quantum computer which quadratically speeds-up every constraint problem and will be thus necessary to solve large problems of this type. We propose to use the remotely-connected Orion system by DWAVE Corporation [50].

1. Introduction.

The research on robot emotions and methods to allow humanoid robots to acquire complex motor skills is recently advancing at a very fast pace [9]. However, assigning simple emotions like “fear” or “anger” or behaviors like obstacle-avoidance to wheeled mobile robots as in Braitenberg Vehicles or subsumption architecture [35,42,43,53], although very useful and of historical importance [10] is practically insufficient to cover all necessary behaviors of future household “helper robots” [11]. Because humans attribute emotions to other humans and to animals, future emotional robots should perhaps be visually similar to humans or animals, otherwise their users would be not able to understand robots’ emotions and correctly communicate with them. Observe that the whole idea of emotional robot helpers is to enable easy communication between humans and robots. Therefore we believe that future emotional robots will be humanoid or at least partially human-like. In our research we concentrate on humanoid robots to express emotions [12]. The research of M. Lukac uses human-like faces and head/neck body combinations. KAIST theatre [13] used whole-body stationary robots with hands. However only a walking biped robot can express the fullness of human emotions by its body gestures, dancing, jumping, gesticulating with hands. Unfortunately larger biped robots are very expensive, in range of hundreds thousands dollars. Fortunately in recent years several small humanoid robots became available for research and entertainment [1 – 7]. We acquired two KHR-1 robots and integrated them to our robot theatre system with its various capabilities such as: sensors, vision, speech recognition and synthesis and Common Robot Language [oo]. OpenCV software from Intel [17] is used for image acquisition and robot vision algorithms. In this paper we would like to share our experiences on the development of the biped robot current status and future projects. A popular approach to solve many motion planning and knowledge-based behavior problems for humanoid robots is the Constraint Satisfaction Model. Unfortunately, for future robots large problems should be solved in real time which will require powerful computers. Observe that while MIT Cog [27] planned to use interaction with environment as a base of learning, it has no walking capability, thus its access to environment is limited. On the other hand the walking robots such as Honda [28] have much developed walking ability giving them access to powerful environmental information, but they lack learning abilities and sophisticated models of environment. Combining both approaches is an ambitious task which can be successful only if large motion-planning/obstacle-avoidance tasks will be executed in real-time and will include machine learning [25,33,38,41,52]. Emotional biped robot exhibits a much broader library of movements and behaviors than a mobile service robot, for instance gesture-related path planning of both hands and the whole body while walking in a room environment is very complicated [48,49]. One way of solving the computer speed problem is to use quantum computers which will give significant speed-up [8,19,51]. Here we propose to use the Orion system from DWAVE Corporation [50] as the first prototype of a quantum computer controlled humanoid robot.

It is shown in this paper how some ideas of quantum computing can be used to build sophisticated robot controllers. It is our hope that the intelligent biped robots will be an excellent medium to teach emotional robotics

[45], robot theatre [13], gait and movement generation, dialog and many other computational intelligence areas that have been not researched yet because of high costs of biped robots. One of the goals of this paper is to help others to start with this new and exciting research area. KHR-1 like robot can become a widely accepted international education platform.

2. KHR-1 Hardware, Assembly and Maintenance.

We purchased two identical kits. The first objective was to make the robot executing what is advertised [1], walking forward and backward, dancing, doing pushups, etc., according to the company-advertised software. This was not a trivial work because all documentation was in Japanese or Korean, and the English translation was done only on our request. Moreover, the kit boxes missed some small components such as screws, washers, and servo hones and we have to disassemble the first robot that was built by a not sufficiently careful and skilled student. If a research group wants to use these kits they should make sure that the person who mechanically assembles the robot is skilled, detail-oriented and is not working in a hurry. Be also sure that all components have been sent to you. Using this kit is not as easy as many other American and European robot kits that we have been using in the past and is definitely not a task for a robotic beginner. In order to ensure that the robot was ready from the hardware perspective, several connections should be checked: (1) The best way to adjust the servo hones is illustrated in Figure 1. The servo hone should be aligned with the middle hole of the cross arm part. (2) The KHR-1 has two servo controllers located on the back of the robot. Each RCB-1 is capable to control up to 12 servos, and they can store data motions designed by the user. Figure 2 shows the two RCB-1 and their connections. Additionally, the Gyroscope is connected between channels 17 and 23, and the Bluetooth is adapted to the serial connection. (3) It is important that the user adjusts screws from time to time during assembly/test. Additionally, the trim function [29] was unable to correct some of the servos. It was necessary to disassemble these servos and realign the splines so they were closer to center. This robot behavior is very sensitive to its assembly and maintenance and a lab assistant with mechanical skills should be delegated to help students. Hopefully good manuals are now available [1 – 6, 17,18,29,46]. Here we mention few points only.

There are certain steps that must be taken to ensure the continued reliability of the robot. First, it is imperative that all the screws attaching the plastic servo discs are present. It was necessary to buy extra 3 millimeter screws from a hobby shop to replace the ones that were missing. As the robot operates, some of these screws will work loose, so it is a good idea to check their tightness periodically. In the future, it is recommended that screws be coated in Locktite brand screw solution to prevent loosening.

3. Motion-related KHR-1 Software

Heart to Heart is the original company software to program and control the KHR-1. The PC interacts with the KHR-1 through the RCB-1 boards which are connected via RS-232 cable. Each board controls the upper and lower body of the robot respectively. The KHR-1 has 17 servo motors. In order to facilitate the programming and controlling of each servo through Heart to Heart software, they have been labeled with numbers as is shown in Figure 3. Each channel shown in the main window of the software represents a specific servo. To illustrate an example, let’s analyze Figure 3 and 4, Channel 6 controls the head, Channel 7 the left shoulder. Be sure that you do not misrepresent numbers and read the assembly and test manual very carefully. We had troubles because of bad translation, but now English manuals can be available from us and perhaps also on the Internet, so the construction and test will be easier for English-speaking robot builders.

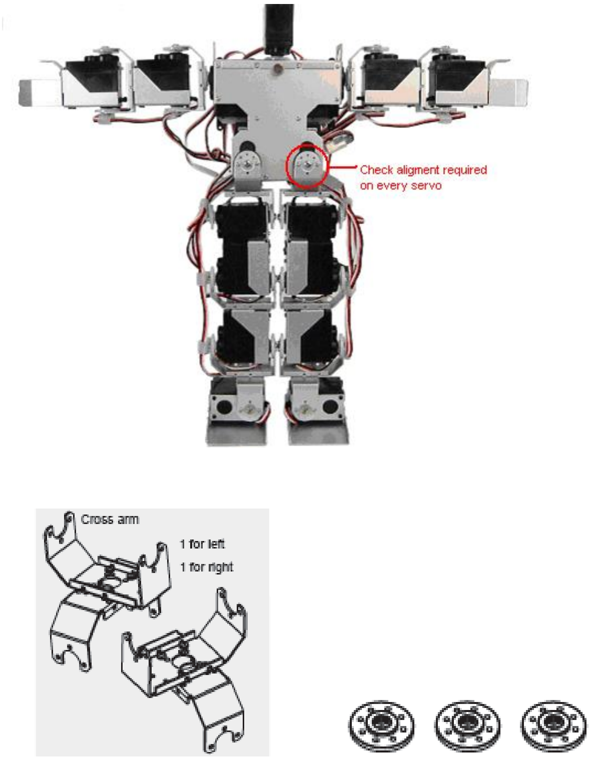


Figure 1. Cross Arms and Servo hones

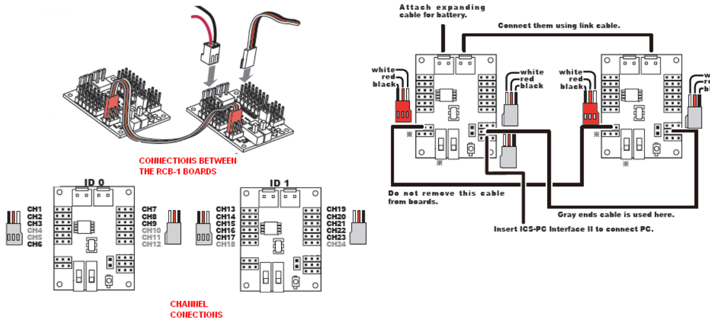


Figure 2.RCB-1s controllers and Servo Cable Arrangements.

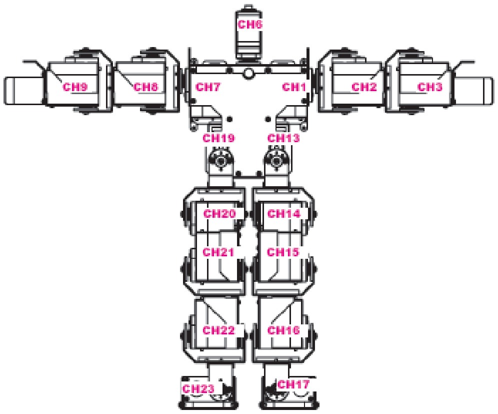
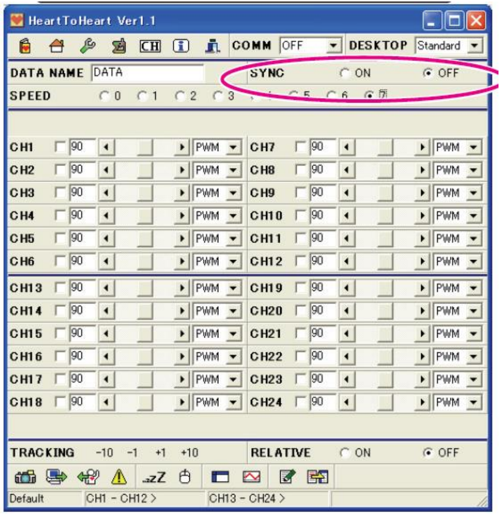


Figure 3. Labeling of the Servo motors.

3.1. SYNC Function The SYNC function (see Figure 4) allows real time communication between the KHR-1 and the Heart to Heart software. When the robot is connected to the PC it is necessary to set the SYNC function in its ON position because it allows to control the robot. If the user wants to make any changes on the servos, create new positions and motion files, the SYNC function must be ON.



COMM OFF DESKTOP Standard

DATA NAME DATA

SYNC ○ ON ● OFF

SPEED ○ 0 ○ 1 ○ 2 ○ 3 ○ 5 ○ 6 ● 7

CH1 ☐	90	PWM	CH7 ☐	90	PWM
CH2 ☐	90	PWM	CH8 ☐	90	PWM
CH3 ☐	90	PWM	CH9 ☐	90	PWM
CH4 ☐	90	PWM	CH10 ☐	90	PWM
CH5 ☐	90	PWM	CH11 ☐	90	PWM
CH6 ☐	90	PWM	CH12 ☐	90	PWM
CH13 ☐	90	PWM	CH19 ☐	90	PWM
CH14 ☐	90	PWM	CH20 ☐	90	PWM
CH15 ☐	90	PWM	CH21 ☐	90	PWM
CH16 ☐	90	PWM	CH22 ☐	90	PWM
CH17 ☐	90	PWM	CH23 ☐	90	PWM
CH18 ☐	90	PWM	CH24 ☐	90	PWM

TRACKING -10 -1 +1 +10 RELATIVE ○ ON ● OFF

Default CH1 - CH12 > CH13 - CH24 >

Figure 4. Heart to Heart Main window.

The Figure 4, shows the first screen that the user gets once the Heart to Heart is opened. The top and bottom bar tool contains important functions that will be explain into detail in the following section. The 24 channels represent each servo motor of the KHR-1. The values displayed represent their position according to their particular center position.

3.2. How to get started. To install the project one needs: HBP files, Visual Basic 6.0 (this is important because you need a "com object."), OpenCV (version 3.1 b). You will also need a version of Visual Basic that supports the com object. We found that VB6 worked well. Access to a supported camera, (we used a Logitech USB web-cam) is also needed. Web-cams are inexpensive and almost any should get you started. It's very important to set all the files up correctly to ensure proper operation. What we provide is a basic setup and you may find better/more advanced options for completing this task. If you are starting from scratch, you will need to generate a method for communicating with the KHR-1 through a com port. That is why it's important to use VB 6.0, later versions do not have this option yet. There is a lot of opportunity to modify and manipulate from this point to take the KHR-1 to the next level! Here our goal is merely to get the ball rolling.

We develop symbolic approach to robot specification based on a Common Robot Language [41]. While the syntax of this language specifies rules for generating sentences, the semantic aspects describe structures for interpretation [34,36]. Every movement is described on many levels, for instance every joint angle or face muscle are at low level and complete movements such as pushups or joyful hand waving are at a high level. These aspects serve to describe interaction with environment at various levels of description. It uses also the constraint satisfaction problem [30,31] creating movements that specify constraints of time, space, motion style and emotional expression. Non-deterministic and probabilistic behaviors are possible within the framework of constraints, allowing more natural behavior of the robot where the movements are logical but not exactly the same in similar environmental or emotional situations. Mechanisms for scripting and scenario writing [44] are also necessary. Humanoid robot movements and emotional behaviors require special notations that take their origins from human emotional gestures and movements such as dances, sport-related and gymnastic movements as well as theatre-related behaviors. These notations and languages originate from choreography, psychology and general analysis of human behavior. Several notations describing human dances exist using Benesh notation, [37,40], LifeForms [39] and others. The goal of our Common Robot Language is to describe human-oriented movements, but it exceeds these behaviors to those like anthropomorphic animals and fairy tale characters.

We created new GUI interface and robot controlling language. There are two main functions that we achieved, the first is mimicking, the second is the behavior state machine.

3.3. Added functions

We focused on new functionality using the command reference from Daniel Albert [3]. Adding new functions and documenting the code where these functions were used will benefit next projects. The next users could look to these as examples of how adopt these functions to program the KHR1. Some of the functions that we added and successfully tested are:

Get home position
Get trim position
Set home position
Set servo trim value

For every function, the value that is returned is a string concatenation of data to be sent to the serial port. The above functions just generate the data the robot expects to see for processing. After receiving the command of interest, the robot then performs the requested operation or sends data back on the communication port.

The ability to read information back from the robot by serial communication was added. The ability to read information doesn't enable any functionality to the objective of mimicking by using video, but the goal was to prepare code for future students such that they could begin using the robot for other applications.

4. Using HBP robot vision software for human mimicking.

OpenCV version 3.1b [17] and the Human Body Project (HBP) software [5] were used in the framework of a state machine to control behaviors mimicked from a human standing in front of the camera. We wanted the KHR-1 to mimic human motion that was being shown on the screen by the HBP software. The HPB works by taking an image of a person's upper body. It then will try and identify the face. Once it can recognize a face it will then look at the body. The image that it acquires is converted to a set of feature (parameters) values assigned to several groups of variables. The variables that we are interested here are the following:
* leftShoulderElevation
* rightShoulderElevation
* leftElbowElevation
* rightElbowElevation

As you can see the values correspond to positions of the joints for each arm.

The openCV software has proven not very responsive to movement and runs poorly on the laptop computer. It is possible that different computer hardware would better run the software or new software would need to be developed. There are many variables in the Human Body Project software that indicate relative position of the eyes, nose, mouth, and arms of the subject. It is definitely possible to use these to make the robot behave in much more complicated fashions. There are many .dll files that the user has to understand the applications of.

One major restriction that we ran into was that the HBP was not a 100% at recognizing the body positions.We found that the robot is very sensitive to non-body objects in the background. We experienced the best performance standing in front of a white wall wearing a dark, solid-color sweater and lit from the front with auxiliary lighting. Even under these conditions, the HBP software recognized body and mouth position correctly only about half the time. Hence, we modified our state machine to respond to gross body movements that were most reliably recognized by the software. This was accomplished by writing a subroutine which tracked the robots arm positions and mouth size. The commands from this state machine were sent to the robot whenever the avatar from the HBP software ran the ShowAvatar routine. Placing a function call to the State Machine function at the end of the ShowAvatar routine provided the trigger mechanism for the state machine function. The state machine code is located in the visual basic project module modKHR1State.

One thing about HBP is that it is slow to respond. Your actions will need to be slow and you will need to hold them until you get the visual feedback from the HBP that it has to see your movement. That is indicated when the avatar moves and holds the new position. (Avatar is a small graphic representation of yourself as a little humanoid as seen by the camera). The HBP is not always accurate. That is something that you'll have to deal with if you don't intend on modifying the original code. That one great thing about HPB, is that you have the option of modifying the original code to some extent and make your own features. To speed up the image recognition we will use the Orion quantum computer in the next project (section 7).

4.1. Interfacing with the KHR-1 controller

We first established what values the HBP software generated for its visual display (the avatar). Based on this we made a translation to transform the values for use with our existing VB/KHR-1 controller. The conversion task was done by taking the output range from the HBP, 50 to

-50 for the elbows and 100 to -100 for the shoulder, and converting it to the output needed for the KHR-1 (0-180) HBP generates four variables that correspond to the right and left elbow angles and the right and left shoulder angle. There were limitations programmed into the VB software that controls the KHR-1 so that the robot would not break a servo by trying to push it's arm into it's body. The values were limited based on the physical constraints of the KHR-1. If both conditions are in that window then we limit the elbow so that it can not hit the body of the robot. Without this function the KHR-1 could hit itself and possibly break a servo.

Understanding your robot's limitations is vital to the success of your project. You may find it useful to manipulate this code to fit your needs, or generate some protective/limiting code yourself. In either case, the better your understanding of the mechanics of your robot, the more success you'll have in controlling it.

5. Gyroscope.

Bipedal humanoid robots are inherently unstable. Unlike wheeled robots, humanoids have a high center of gravity and must balance carefully in order not to tip over as they move. While it is possible to achieve balance in the absence of feedback sensors, slight variations in the environment often cause imbalance and result in a fall. Several approaches have been taken to improve the stability of two legged robots. Installation of large foot pads aid in stability, but can be cumbersome in quick maneuvers.

One way to improve stability without adding area to the feet of the robot is to add a feedback mechanism. Feedback is present in many natural and man-made systems. The principle of negative feedback and control theory has been instrumental in achieving reliability in mechanical and electrical systems. In order to improve the stability of the bipedal robot, a compensating gyroscope was installed. This unit was manufactured by the Kondo company, and was designed specifically for the KHR-1. Thus, it was trivial to simply plug the gyroscope into the cabling without modification of wiring. The gyroscope works as follows: Each servo motor receives a pulse width modulated (PCM) square wave signal from the controller board on the robot. The controller board encodes position commands to each servo motor by modifying the duty cycle of the PCM input. The gyroscope is wired in series with the servo motors to be controlled. That is to say that the PCM signal passes through the gyroscope wherein the duty cycle is modified according to the instantaneous acceleration in the axis to which the gyroscope is sensitized. This has the effect that sudden acceleration will result in compensatory movement of the servos to correct and maintain balance.

The gyroscope installed on this robot is sensitive to acceleration in only one of two possible corrective axes. One pair of servos controls side to side balance at the base of the feet. Another can provide front to back correction by changing the angle of bend at the knee joints in the legs. It would be necessary to have two separate gyroscopes to provide balance feedback for both front to back and side to side motions.

We have only one gyroscope, and chose to control side to side balance. Our choice for side to side motion was due to the fact that additional hardware is necessary to program the servos 22 and 16. According to the translated instructions, the "Servo Manager" application along with the special cable available from robosavvy.com is necessary to program servos 22 and 16 to be able to accept the signal from the gyroscope. This is in contrast with the software-free modification of the side to side axis. In any case, installing the gyro helped with movement stability and we plan to add also the second gyro.

6. Constraint Satisfaction for Emotional Robotics

Based on our experience and also on literature, one weakness of current robots is insufficient speed of robot image processing and pattern recognition. This can be solved by special processors, DSP processors, FPGA architectures and parallel computing. We applied already these approaches in our past research. The trouble is that designing or programming many partial processing algorithms is very time consuming. On the other hand, logic programming language such as Prolog allows to write all kind of such programs very quickly, but the software is not efficient enough. An interesting approach is to formulate many problems using the same general model. This model may be predicate calculus, Satisfiability, Artificial Neural Nets or Constraints Satisfaction Model. Many problems, for instance the well-known Waltz algorithm can be reduced to it.

Huffman and Clowes created an approach to polyhedral scene analysis, scenes with opaque, trihedral solids, next improved significantly by Waltz [56], which popularized the concept of constraints satisfaction and its use in problem solving, especially image interpretation. Objects in this approach had always three plane surfaces intersecting in every vertex. Thus there are 18 possible trihedral vertices in this problem out of 64 possible. There are only 3 types of edges between these blocks possible: (1) obscuring edge is a boundary between objects or objects and background. Boundary lines are found using outlines with no outside vertices, (2) concave edges are edges between two object's faces forming an acute angle when seen from outside, (3) convex edges are those between two faces of an object forming an obtuse angle as seen from outside. There are only four ways to label a line in this blocks world model. The line can be convex, concave, a boundary line facing up and a boundary line facing down (left, or right). The direction of the boundary line depends on the side of the line corresponding to the face of the causing it object. Waltz created a famous algorithm which for this world model which always finds the unique correct labeling if a figure is correct. Moreover, the algorithm handled also shadows and cracks in blocks. Mackworth and Sugihara extended this work to arbitrary polyhedra and Malik to smooth curved objects. This becomes a well-known approach to image recognition based on constraint satisfaction and a prototype of many similar approaches to vision and planning problems in robotics.

Constraint satisfaction model is one of few fundamental models used in robotics [57,58,59,60,61,62,63]. It is used in main areas of robotics and especially in vision, knowledge acquisition, knowledge usage including in particular the following: planning, scheduling, allocation, motion planning, gesture planning, assembly planning, graph problems including graph coloring, graph matching, floor-plan design, temporal reasoning, spatial and temporal planning, assignment and mapping problems, resource allocation in AI, combined planning and scheduling, arc and path consistency, general matching problems, belief maintenance, experiment planning, satisfiability and Boolean/mixed equation solving, machine design and manufacturing, diagnostic reasoning, qualitative and symbolic reasoning, decision support, computational linguistics, hardware design and verification, configuration, real-time systems, and robot planning, implementation of non-conflicting sensor systems, man-robot and robot-robot communication systems and protocols, contingency-tolerant motion control, multi-robot motion planning, multi-robot task planning and scheduling, coordination of a group of robots, and many others.

7. Adiabatic Quantum Computing to solve Constraint Satisfaction Problem efficiently.

It is quite possible that the date of February 13th 2007 will be remembered in annals of computing. DWAVE company demonstrated their Orion quantum computing system in Computer History Museum in Mountain View, California. It was the first time in history that a commercial quantum computer was presented. The Orion system is a hardware accelerator designed to solve in principle a particular NP-complete problem called the two-dimensional Ising model in a magnetic field (for instance quadratic programming). It is built around a 16-qubit superconducting adiabatic quantum computer (AQC) processor. The system is designed to be used together with a conventional front end for any application that requires the solution of an NP-complete problem. The first application that was demonstrated was pattern matching applied to searching databases of molecules. The second was a planning/scheduling application for assigning people to seats subject to constraints. This is an example of applying Orion to constraint satisfaction problems. Other problems of this type include graph coloring, maximum clique and maximum independent set. Yet another class are SAT (satisfiability) problems. As we know, many of these problems, the constraint-satisfaction problems are important components of robotic software. The company promises to provide free access by Internet to one of their systems to those researchers who want to develop their own applications.

The plans are that by the end of year 2008 the Orion systems will be scaled to more than 1000 qubits. It is even more amazing that the company plans to build in 2009 processors specifically designed for quantum simulation, which represents a big commercial opportunity. These problems include protein folding, drug design and many other in chemistry, biology and material science. Thus the company claims to dominate enormous markets of NP-complete problems and quantum simulation. If successful, the arrival of adiabatic quantum computers will create a need for the development of new algorithms and adaptations of existing search algorithms (quantum or not) for the DWAVE architecture. The arrival of Orion systems is certainly an excellent news for any research group that is interested in formulating problems to be solved on a quantum computer. In this project we plan to concentrate on robotic applications of the Constraint Satisfaction Model.

Adiabatic Quantum Computing was proved equivalent [47,55] to standard QC circuit model that we used in [20 – 26], thus at least in theory each of the developed by us methods can be transformed to an adiabatic quantum program and run on Orion. We developed logic minimization methods to reduce the graph that is created in AQC to program problems such as Maximum Clique or SAT. This programming is like on "assembly level" or "machine language" but with time more efficient methods will be developed in our group. This is also similar to programming current Field-Programmable Gate Arrays. The processor is programmable for a particular graph abstracting the problem. We predict that in future adaptations of many methods developed for FPGAs will be used for quantum computers.

Several aspects presented below will be considered while creating software for the Orion AQC:

1. One method of creating software for AQC is by formulating an oracle for Grover algorithm and next converting it to the AQC model [47,55]. This requires the ability to synthesize a complex permutative circuit (reversible circuit) from universal binary gates such as Toffoli or Fredkin. Adiabatic equivalent of Grover algorithm is implemented in Orion system and 16-qubit oracles can be built for Orion system. This is not enough for larger problems, but it is a good starting point for self-education. The developed by us minimization methods [24] can be used to synthesize complete oracles or their parts, for incomplete functions.

2. To practically design oracles for Grover as quantum circuits one has first to formulate various NP-complete problems and NP-hard problems as oracles. Some robotic problems, especially in vision (such as convolution, matching, applications of Quantum Fourier Transform and other spectral transforms [4,5,17,32,56,57,58]) require quantum circuits that are not permutative but use truly quantum primitives like the controlled phase gate. Methods to convert these circuits to AQC model should be investigated and the problems should be converted to AQC model and executed on Orion.

3. We proposed an algorithm to find the best polarity Fixed-Polarity-Reed-Muller transform [20]. This can be used as a machine learning method when a function with don't cares is given at the inputs. Similarly the method presented in [24] is a general purpose machine learning method from examples. Next, Quantum Neural Network can be synthesized. In a non-published research we extended Quantum Fourier Transform based convolution/matching methods to Haar, complex Hadamard and other spectral transforms. Several image processing algorithms can be created for quantum computers with significant complexity reduction [57,58]. These algorithms use not only constraint satisfaction, SAT and search but also quantum spectral transforms and solving general purpose Schroedinger equations.

4. We work also on SAT, maximum clique, Hamiltonian Path, shortest path, travelling salesman, Euler Path, exact ESOP minimization, maximum independent set, general constraint satisfaction problems such as cryptographic puzzles, and other unate/binate/even-odd covering problems, non-Boolean SAT solvers and equation-solvers. For all these problems we built oracles and we plan to convert them to AQC.

5. Development of new quantum algorithms based on extensions and adaptations of Grover, Hogg and other quantum search and Quantum Computational Intelligence models. Generalizations of Grover, Simon and Fourier transforms to multiple-valued quantum logic [19,21,22,23] as implemented in the circuit model of quantum computing. Analysis and comparison with binary quantum algorithms and their circuits. Conversion to AQC model.

6. Generalizing well-known quantum algorithms to multiple-valued quantum logic. For instance, in paper [23] we generalized the historically famous algorithm by Deutsch and Jozsa to arbitrary radix and we proved that affine functions can be distinquished in a single measurement. Moreover, functions that can be described as "affine with noise" can be also distinguished. This can be used for very fast texture recognition in robot vision. We work also on generalization of Grover to multiple-valued quantum circuits.

7. All these problems are useful in robotics to solve various vision and pattern recognition path-planning, obstacle avoidance and motion generation problems. Observe that every NP-complete problem can be reduced to Grover algorithm and Grover reduced to AQC model that can be run on Orion. Similarly the classes of quantum simulation algorithms will be run of future DWAVE architectures. Although the speedup of the first of the classes is only quadratic, it will be still a dramatic improvement over current computers. It is also well-known that if some heuristics are known for an NP-complete problem, one of several extensions and generalizations to Grover can be used, which may provide better than quadratic speedup, but is problem-dependent. Since however all classical solvers of NP-Complete problems that are used now in industry are heuristic and better than their exact versions, we believe that the same will happen when quantum programming will become more advanced.

The work presented here in the framework of "Quantum Robotics" is new. It is different than "quantum robots" proposed by Benioff [54] where robot operates in structured quantum environment rather than in standard mechanics environment, or the work from [14] which is limited to one aspect of mobile robotics only. However, our model of a quantum robot, which may use quantum sensors but operates on normal effectors in standard environment is closer to the model from [14] than the one from [54]. Our model of a quantum robot applies quantum concepts to sensing, planning, learning, knowledge storing, general architecture and movement / behavior generation. [8,25,41]. It uses quantum mappings as in [53,42], quantum automata [42], Deutsch-Jozsa-based texture recognition [23], Grover-based image processing, emotional behaviors [12], quantum learning

[13,24,25,52] and motion planning and spectral transforms as its special cases.

8. Conclusions and future work.

As seen on the video, KHR-1 is now able to mimic upper body human motions. The software and videos are available on Marek Perkowski's Webpage. Students who work on this project learn about robot kinematics, robot vision, state machines (deterministic, non-deterministic, probabilistic and quantum - entangled) robot software programming and commercial robot movement editors. The most important lesson learned is the integration of a non-trivial large system and the appreciation of what is a real-time programming. It is important that the students learn to develop a "trial and error" attitude and also how to survive using a non-perfect and incomplete documentation. It was also emphasized by the professor that students create a very good documentation of their work for the next students to use [2,18]. The student team spent many hours trying to improve the motion files for walking, turning, standing up and other leg-related movements. Whereas it is easy to teach the robot to dance with the upper body, it proved frustrating to involve the legs of the robot in any motion command. Finally few safe leg movements were developed but further work using more foot sensors and more advanced movement generation software appears neccessary. The motion files of the robot need to be better defined and more of their variants should be created. This will probably best be done with a genetic algorithm, but will require either human or computer vision feedback to judge the success of any particular algorithm for a motion. Future teams would be well advised to become well familiar with the motion teaching method early in the project to save time and avoid hurried effort at the class end.

In the second research direction the interface to Orion system will be learned and how to formulate front-end formulations for various robotic problems as constraint-satisfaction problems for this system.

REFERENCES

1. KHR-1 robot company. KHR-1 Hardware Manual"
2. Q. Williams and S. Bogner, 478 KHR-1 Final Report, 2006.
3. Daniel Albert, RCB 1st draft, a command outline, private communication.
4. Human Body Project Webpage. http://www.fuzzgun.btinternet.co.uk/rodney/components.html
5. Human Body Project Webpage. https://sourceforge.net/projects/hbp/
6. Robosavvy company Webpage. http://www.robosavvy.com/
7. http://www.sparkfun.com/
8. M. Perkowski, *Quantum Robotics for Teenagers*, book in preparation. 2007.
9. C. Breazeal, *Designing Sociable Robots*,. MIT Press, 2002.
10. V. Braitenberg, *Vehicles: Experiments in Synthetic Psychology*. MIT Press. 1986.
11. A.Green, H.Huttenrauch, M.Norman, L.Oestreicher, K.Severinson Eklundh, User Centered Design for Intelligent Service Robots, *Proc. Intern. 2000 Workshop on Robot and Human Interactive Communication*, Osaka, Japan, September 27-29.
12. M. Lukac and M. Perkowski, "Quantum mechanical model of emotional robot behaviors," in *Proceedings of the ISMVL 2007*, 2007.
13. M. Perkowski, T. Sasao, J-H. Kim, M. Lukac, J. Allen, S. Gebauer, Hahoe KAIST Robot Theatre: Learning Rules of Interactive Robot Behavior as a Multiple-Valued Logic Synthesis Problem, *Proc. ISMVL 2005*, pp. 236-248.
14. D. Dong, Ch. Chen, Ch. Zhang, and Z. Chen, An Autonomous Mobile Robot Based on Quantum Algorithm, pp. 393-398.
15. Z. Tang, Ch. Zhou, Z. Sun, Humanoid Walking Gait Optimization Using GA-based Neural Network, *ICNC* (2) 2005, pp. 252-261.
16. S. Zhou, Z. Sun, A New Approach Belonging to EDAs: Quantum-Inspired Genetic Algorithm with Only One Chromosome, *ICNC* (3) 2005, pp. 141-150.
17. OpenCV http://www.intel.com/technology/computing/opencv/
18. Q.Williams, S. Bogner, M. Kelley and C. Castillo, KHR-1 and the HBP Interface. Manual for installing, running and interfacing the Human Body Project (HBP) with KHR-1 for mimicking and control. *Users' Manual. ECE PSU*, 2006.
19. M.Perkowski, "Multiple-Valued Quantum Circuits and Research Challenges for Logic Design and Computational Intelligence Communities," *Invited Paper*, *IEEE ConneCtIonS*, IEEE Computer Intelligence Society, November 2005, pp. 6-12.
20. Lun Li, M. A. Thornton, M.A. Perkowski, "A Quantum CAD Accelerator Based on Grover's Algorithm for Finding the Minimum Fixed Polarity Reed-Muller Form," *Proc. ISMVL 2006*, pp. 33.
21. M.H.A. Khan and M. Perkowski, "Quantum Realization of Ternary Parallel Adder/Subtractor with Look-Ahead Carry," *Proc. International Symposium on Representations and Methodologies for Emergent Computing Technologies*, Tokyo, Japan, September 2005. pp. 15-22.
22. M.H.A. Khan, and M.Perkowski, "Quantum Realization of Ternary Encoder and Decoder," *Proc.*

International Symposium on Representations and Methodologies for Emergent Computing Technologies, Tokyo, Japan, September 2005. pp. 23 – 27.

23. Y. Fan,, Generalization of Deutsch-Jozsa algorithm to Multiple-Valued Quantum Logic, *Proc. ISMVL* 2007.
24. M. Kumar, B. Year, N. Metzger, Y. Wang and M. Perkowski, Realization of Incompletely Specified Functions in Minimized Reversible Circuits. *Submitted to RM* 2007.
25. M. Lukac and M. Perkowski, “Quantum behaviors: Measurement and synthesis,” in Submitted to *Reed-Muller 2007*, 2007.
26. N. Giesecke, Dong Hwa Kim, S. Hossain and M.Perkowski, Search for Universal Ternary Quantum Gate Sets with Exact Minimum Costs. *Submitted to RM* 2007.
27. R. A. Brooks, C. Breazeal, M. Marjanovic, B. Scassellati, and M. M. Williamson, “The Cog Project: Building a Humanoid Robot,” in *IARP First International Workshop on Humanoid and Human Friendly Robotics,* (Tsukuba, Japan), pp. I- 1, October 26-27 1998.
28. Honda Humanoid Robot Project http://world.honda.com/robot/
29. Q. Williams and S. Bogner, The ECE478 final report.
30. V. Kumar, Algorithms for constraint-satisfaction problems: A survey. *AI Magazine*, 13 (1): 32-44, 1992.
31. A. Mackworth, Consistency in networks of relations, *Artificial Intelligence*, 8(1): 99-118, 1977.
32. A.K.C. Wong, S.W. Lu, and M. Rioux, Recognition and shape synthesis of 3D objects based on attributed hypergraphs, *IEEE Trans. on Pattern Anal. and Mach. Intell*, 11:279-290, 1989.
33. Badler, N. I., Bindiganavale, R., Granieri, J. P., Wei, S., and Zhao, X. “Posture Interpolation with Collision Avoidance,” In *Proc. Computer Animation*, pp13-20, 1994.
34. Badler, N. I., and Smoliar, S. W. “Digital Representation of Human Movement”, *Computer Surveys*, Vol. 11, No 1, March 1979.
35. D.-H. Kim, Ch. Brawn, M. Sajkowski, T. Stenzel, T.Sasao, J. Allen, M. Lukac, T. Wang and M. Perkowski, Artificial Immune-Neuro-Fuzzy System to control a walking robot Hexor, *Proc. ULSI* 2006.
36. T.W. Calvert, Armin Bruderlin, Sang Mah, Thecla Schiphorst, and Chris Welman, “The Evolution of an Interface for Choreographers”, *Interchi*, pp. 24-29, 1993.
37. M. Causley, *An introduction to Benesh Movement Notation*, ISBN: 0836992806, 1980.
38. B. Choi, “Automata for Learning Sequential Tasks,” *New Generation Computing: Computing Paradigms and Computational Intelligence*, Vol. 16, No. 1, pp. 23-54, 1998.
39. Schiphorst, T. “LifeForms: Design Tools for Choreography”, *Dance and Technology I: Moving Toward the Future*, pp. 46-52, 1992.
40. R. Ryman, B. Singh, J. Beatty and K. Booth, “A Computerized Editor of Benesh Movement Notation,” *Dance Research Journal*, 16(1): 27-34, 1984.
41. M. Lukac, Robots, Emotions, Incompleteness and Quantum Computing, *Ph.D. Thesis in preparation*, PSU,2006.
42. A. Raghuvanshi, Y. Fan, M. Woyke and M. Perkowski, Quantum Robots for Teenagers, *Proc. ISMVL* 2007.
43. R. A. Brooks, “Intelligence without reason,” in *Proceedings of the IJCAI*, 991, pp. 569–595.
44. K. Perlin, A.Gikdberg, “Improv: A System for Scripting Interactive Actors in Virtual Worlds”, *Computer Graphics Proceeding*, pp.205-216, 1996.
45. C. Breazeal and B. Scassellati, “How to build robots that make friends nd influence people,” in *Proceedings of IROS*, 1999, pp. 858–863.
46. Robotic class 2006. PSU_KHR1_Interface.pdf
47. D. Aharonov and A. Ta-Shma, Adiabatic Quantum State Generation and Statistical Zero Knowledge, *Proceedings of the 35th Annual ACM Symposium on Theory of Computing.* ACM Press, New York, 2003, pp. 20–29.
48. H. Lim, A. Ishii, and A. Takanshi, “Basic motional walking using a biped humanoid robot,” in *Proceedings of the IEEE SMC*, 1999.
49. T. Nakata, T. Sato, and T. Mori, “Expression of emotion and intention by robot body movement,” in *Proceedings of the 5th International Conference on Autonomous Systems*, 1998.
50. `http://dwave.wordpress.com/2007/01/19/quantum-computing-demo-announcement/`. Look also to many materials linked from this webpage.
51. M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information*. Cambridge University Press, 2000.
52. M. Lukac, M. Perkowski, H. Goi, M. Pivtoraiko, Ch-H. Yu, K. Chung, H. Jee, B.G. Kim, and Y-D. Kim, Evolutionary approach to Quantum and Reversible Circuits synthesis, *Artificial Intelligence Review Journal, Special Issue on Artificial Intelligence in Logic Design, S.Yanushkevich guest editor*, 2003.
53. Ch. Brawn, N. Metzger, J. Biamonte, M. Lukac, A. Aulakh, I. Devanath, M. Sajkowski, T. Stenzel, D.H. Kim, T. Sasao and M. Perkowski, Hexor, a Walking and Talking Robot with Quantum and Fuzzy Inference, *Proc. ULSI* 2006.
54. P. Benioff, Quantum Robots and Environments, Phys. Rev. A 58, pp.893–904, Issue 2, August 1998.

55. A. Mizel, D. Lidar, M. Mitchell, Simple proof of equivalence between adiabatic quantum computation and the circuit model. *2007 APS March Meeting*, Denver, Colorado, http://www.aps.org/meeting/march
56. D.L. Waltz, Understanding Line Drawings of Scenes with Shadows in P. H. Winston ed. Psychology of Computer Vision pp.19-91, *McGraw-Hill*, New York.1975.
57. D. Curtis, and D.A. Meyer, Towards quantum template matching, preprint.
58. G. Beach, Ch. Lomont, Ch. Cohen, Quantum Image Processing, *Proc. 32nd Applied Imagery Patter Recognition Workshop (AIPR'03)*, p. 39.
59. S. Minton and M. D. Johnston and A. B. Philips and P. Laird, Solving Large-Scale Constraint Satisfaction and Scheduling Problems Using a Heuristic Repair Method 1990, *Proc. of AAAI-90*, Boston, MA", pp. 17-24.
60. M. P.J. Fromherz, T. Hogg, Yi Shang, Modular Robot Control and Continuous Constraint Satisfaction, *Proc. IJCAI-01 Workshop on Modelling and Solving Problems with Constraints*, Aug. 2001.
61. Q. Huang, K. Yokoi, S. Kajita, K. Kaneko,H. Arai, N. Koyachi, and K.Tanie, Planning Walking Patterns for a Biped Robot, *IEEE Trans. On Robotics and Automation*, Vol. 17, No. 3, June 2001.
62. S. Gualandi, B. Tranchero, Concurrent constraint programming-based path planning for uninhabited air vehicles, preprint.
63. D.K. Pai, R. Barman, Constraint Programming for Platonic Beast Legged Robots, *Proc. Intern. Conf. on Robotics and Automation*, Minneapolis, 1996.