

# Distributed Control System on a model of Mars Rover Spirit

Rerngwut Choomuang, Chiroth Charitkhuan, Janjai Bhuripanyo  
 Department of Mechatronics & Department of Computer Engineering  
 Sripatum University (SPU)  
 Bangkok, Thailand  
[rerngwut.ch@spu.ac.th](mailto:rerngwut.ch@spu.ac.th), [chiroth.ch@spu.ac.th](mailto:chiroth.ch@spu.ac.th), [janjai.bh@spu.ac.th](mailto:janjai.bh@spu.ac.th)

**Abstract**—The achievement in VLSI technology has changed the old style single CPU integration into new concept of Distributed Control on multiple controllers which emphasis on the data communication between controllers. The main controller sends command and retrieves data/status from the slave controller(s) and the slave controller(s) act accordingly, as the result, the software development cycle has decreased drastically. The paper is an example of an effort to implement Distributed Control concept into a model of Mars Rover Spirit which in turn simplifies the design and allows the designer to easily add more software features to the robot in the future.

**Keywords**—Distributed Control, Mars Rover Spirit

## I. INTRODUCTION

THE concept of Distributed Control was originated way back in the 1970+ in USA and Europe. The manufacturing industries in USA were good examples on utilizing the Distributed Control, various manufacturers had slowly converted most or all of the existing manual machines into NC (Numerical Control) and CNC (Computer Numerical Control) machines (for examples: Machining Center, Lathe, CMM (Coordinating Measurement Machine) machines, etc). Since most of the early machines were built as stand-alone machines which in turn likely to created bottle-necks and many other related problems in the manufacturing lines later on. Technologies such as: Machining Cell, DNC (Direct Numerical Control), FMS (Flexible Manufacturing System) and CIM (Computer Integrated Manufacturing) were introduced respectively from the late 1970 to the later part of 1990+, all of which aimed at the ultimate control over the whole manufacturing processes (the just-in-time inventories, on-time deliveries, etc) to keep up with the rest of processes in the business world. The transition of the above revolution had accepted the concept of Distributed Control in the manufacturing processes as an important part of their successes.

A bull race condition in the VLSI technology which started in the early part of 1970+ and continued until today, had drastically changed old style thinking and opened to

new and untouched opportunities throughout the world. The concept of Distributed Control in the manufacturing industry can now be implemented and tested with the integration of low cost microcontrollers at various research centers and laboratories in most countries. The concept of Distributed Control has been extended to the rest of the industries, such as, automotive industry, electrical and appliances industry, communication industry and many others. It is very common in the industry today to see group of controllers work together to accomplish the required task rather than trying to put everything on single CPU. The industry already proved and accepted the Distributed Control concept and the result of this paper will support such believes.

When to apply the Distributed Control concept depends on the project itself, the single CPU approach is not appropriate for the proposed Mars Rover Spirit project due to the following reasons:

- Hardware design is complicated. Since there are at least 42 motors to be controlled and each of them requires up to 6 I/O control lines to properly control each motor. Two output lines for the operation selection on motor (such as: CW rotation, CCW rotation and Stop), two input lines for encoder feedback (feedback counts and direction of rotation), two input lines for 2 limit switches (one on each side of the motor rotation). At least a total of  $42 \times 6 = 252$  control lines are required for the proposed Mars Rover Spirit. Such requirements make the designer adds more parallel I/O chips to the design on the single CPU approach. Normally, a parallel I/O has up to 24 control lines but some of them doesn't has the bit assignment capabilities, this force the designer to spread out the 6 control lines over several I/O ports which in turn create more hardware complication.
- Limited interrupt lines on CPU. It is very likely that standard CPU has up to 2 interrupt lines. The feedback counts from the encoder feedback circuit must be tied to the interrupt line of the CPU to get the accurate feedback counts and there are up to 42 feedback counts (each of them from each motor). Even though the hardware designer can overcome the

above obstacle, with the single CPU approach, the CPU will spend most of her time servicing all of the above incoming interrupts. Since there are limited interrupt lines on the CPU, the interrupt polling approach must be implemented in the software to find the cause of the incoming interrupt(s), this will slow down the interrupt response(s) on the single CPU approach drastically.

- Communication problems. There are also serial commands between the PC and the above CPU. Depending on the protocol, the number of bytes in the protocol can vary from a simple 8 bytes (if such command handles 2 motors operation simultaneously) to a rather lengthy command for all 42 motor operations on the single CPU approach. There are also the status response(s) that needs to send back to PC from the above CPU. The communication in both cases are likely to use interrupt on the CPU's serial port, this will definitely put additional burden on the single CPU approach.
- Complex software implementation. A total of 42 DC motors are being used, additional PWM software must be developed for all of the 42 motors. Even though some CPUs (such as AT89C51RD2) may have additional PWM hardware built in but they are still not enough for all the 42 motors in this project. All of the above requirements make software more complicate and difficult to develop. Most of the system cost now a day is on the software side, such complicated software requires more time and budget.
- Additional unforeseeable problems. For most of any development cycles, it is very common to encounter many unexpected problems, for this task, there will be more hidden problems behind the scenes and need to be solved as project moves along.

With the above limitations, the concept of Distributed Control will be utilized in this project instead. For this project, the Distributed Control has been carefully planned and applied to build a scaled down version of the NASA Mars Rover Spirit which landed on Mars in 2001.



Fig 1. A Graphic model of NASA Mars Rover Spirit

A team consists of 6 undergraduate students from the Department of Computer Engineering, they are under the supervision from 3 advisors (two from the Department of Computer Engineering and one from the Department of Mechatronics at Sripatum University).

The goal of this project is to prove that with the concept of Distributed Control when properly planned and executed will simplify the design, create parallel development processes, manageable testing and integration, and make the final system test(s) within a relatively reasonable of time and budgets.

## II. RELATED WORK

Distributed Control architecture has been around for years especially in the US industries. In research area, there are many related papers regarding to this architecture, the number of controllers in each system varied depending on the interested research papers, examples of them are [1], [2], [3], [4]. The number of 24 controllers (including 2 PC) integrated in this project is by far greater than those of the above researches.

For the information on the NASA Mars Rover Spirit and their respective specifications, they are still top secret at NASA and will not be released to public in any foreseeable future. The only piece of public document being released on the NASA Mars Rover Spirit is only a 25 minutes of simulated video which can be downloaded from NASA's website [5] and the news from other press releases on the various Mars operations.

## III. PROPOSED MODEL OF MARS ROVER SPIRIT

An extensive review on the Mars Rover Spirit's video [5] from NASA has been done to identify all the necessary information and functions performed on the Mars Rover Spirit. Additional searches were required to obtain the rest of the missing data such as: actual dimension of the Mars Rover Spirit, etc. Various group meetings were initiated to identify the scope and the specification of each individual parts of the project, what is the limitation, all the necessary hardware (both mechanical hardware and control hardware), software, etc. The specification which included the processes (main process, sub-processes which had to be done separately or parallel), design, build, test, integration and system test, etc. have been drawn and a proposal has been made to obtain all the necessary funding from the university. The following is a typical specification on the proposed model of Mars Rover Spirit, they are as follow:

- 1) A proposed model is approximately 2/5 the actual size of the NASA Mars Rover Spirit
- 2) All the driving mechanism of the proposed model will be similar to the original Mars Rover Spirit in functions due to limitation of parts available locally.
- 3) The expansion cycle of the proposed model is also similar but the initial expansion cycle of all the wheels (after they were packed at NASA before launching into

space) which is too complicate and difficult to design at this stage has been eliminated.

- 4) The arm of the proposed model will be modified from the original Mars Rover Spirit to a two fingers gripper to allow the proposed model to pickup and move specific object(s).
- 5) Solar cell panel will be equipped on the proposed model as much as possible including all the necessary charging circuit to charge the batteries on the proposed model.
- 6) The Distributed Control concept will be implemented to simplify the design of control hardware and software. A RS-485 serial network of 24 microcontrollers (Atmel AT89C2051) will be created. This network allows 22 of the controllers perform various motor control functions and two of controllers control the vision system.
- 7) Three software programs will be developed, two of them to support the Distributed Concept on the proposed model and the third ones is on an external PC. This external PC will act as system console for the proposed model both the movement and vision.
- 8) The proposed model will have additional teaching capabilities which allow the proposed model to be taught, for example, teach the proposed model to move to a table, pick up a can, move and place the above can at the required destination. Once the teaching processes have been completed, the proposed model should be able to repeat the whole sequences by itself on one single command.

#### IV. PHASE I (HARDWARE DESIGN)

The hardware design team has been broken down into 2 smaller groups, one group concentrated on the mechanical design which worked under very close supervision of the advisor from the Department of Mechatronics. The CAD program such as: SolidWork has been utilized to create the proposed model of 2/5 of the actual NASA Mars Rover Spirit. Repeated modifications have been done to change the original design over and over again due to the availability of the 2<sup>nd</sup> hand parts in the local market. Even though the proposed model started to look different from the original design but the functionality still the same. Simulation features available in the SolidWork software helped the design team to spot and correct any obvious failure points on the proposed model during the design phase. Fig.2 and Fig.3 are typical simulation made on wing and wheel during the design phase respectively. Fig. 4 shows the typical differences between the actual design and the original of the NASA Mars Rover Spirit. Fig. 5 is the proposed model of NASA Mars Rover Spirit.

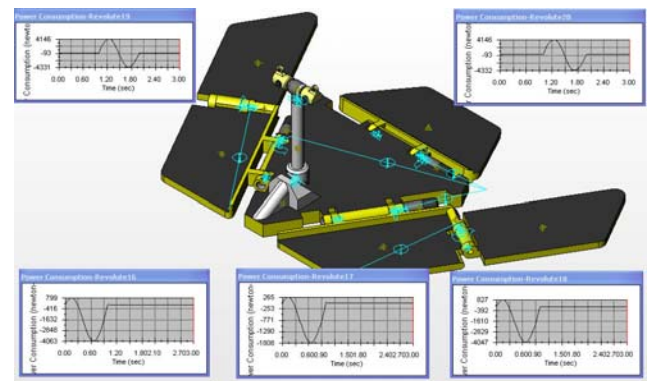


Fig.2. A typical simulation on wing during the design phase.

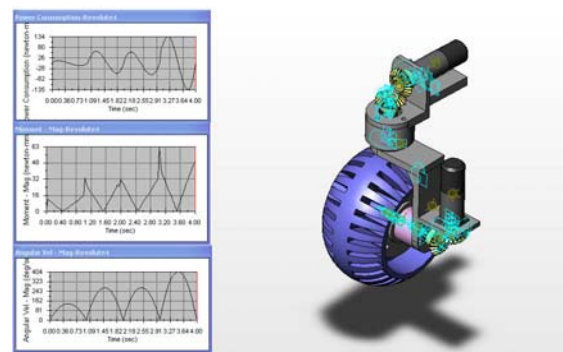


Fig.3. A typical simulation on wheel during the design phase



Fig.4. Typical modified wheel against the original wheel



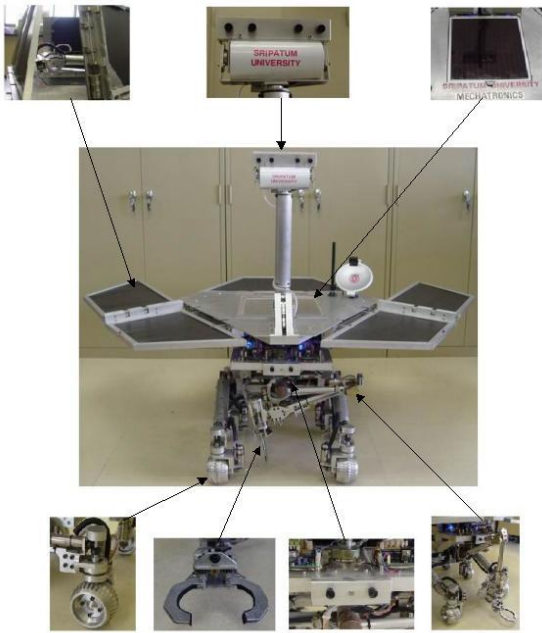


Fig.5. Proposed model for NASA Mars Rover Spirit

The second group concentrated on the hardware control units of the proposed model. At least 2 version of hardware (all of which based on the Atmel AT89C2051) have been designed and tested. There were minor modifications on the first design, the hardware was tested and a small assembly language program was written for both the open and closed loops to control three different sets of motors. The driver circuit was carefully checked to make sure that they could handle types all of the above motors. The design team decided to use the same hardware for all motors to keep the number of hardware and spare parts down for easy maintenances later on. The first version worked relatively well under all test conditions but one major drawback which forced the design team to abandon the first hardware version was the wiring complication occurred during the first integration attempt. Limited space available on the proposed model made the wiring, checking and testing very difficult.

After 2 weeks on the first integration attempt, the design team was forced to abandon the first version and started to work on the 2<sup>nd</sup> version of hardware. All the unnecessary wiring were eliminated and resided on the PCB (which housed a total of 11 units of Atmel AT89C2051 per board), only the necessary wiring were allowed to wire on the proposed model. The 2<sup>nd</sup> hardware version had only one minor modification on the PCB board and the board itself was up and running within a few days. Two more boards were populated and tested, only two boards were necessary for the proposed model and the third one was kept as spare part. The purpose of this PCB board was to control the motors at various locations on the proposed model of Mars Rover Spirit. One particular AT89C2051 on the PCB board can control up to 2 motors simultaneously and two boards can control up to  $11 \times 2 \times 2 = 44$  motors altogether. Two PCB boards on the 1<sup>st</sup> version were brought back to handle the

vision system which consisted of 7 cameras on the proposed model (4 on top (head) of the robot, 2 in the middle-front side of the robot and the last one on the gripper of the robot).

Additional PC notebook was added to act as overall controller on the proposed model and provided a wireless communication between the proposed model and external PC. Fig.6. (a) and (b) is a typical control on the proposed model of NASA Mars Rover Spirit.

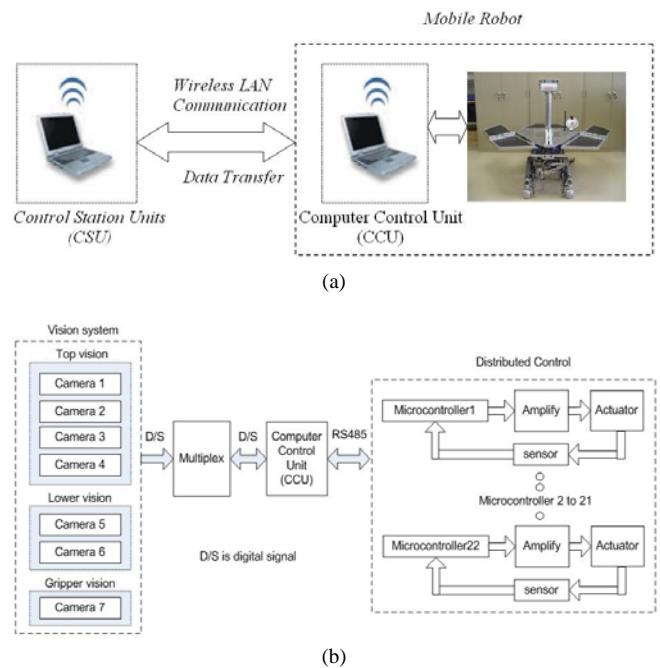


Fig.6. Typical control on the proposed model

## V. PHASE II (SOFTWARE DESIGN)

According to the initial plan, there were 3 separated software programs that needed to be developed, they are as follow: software on an external PC (as system console software), software on a notebook PC on the proposed model and the software on the Atmel AT89C2051. Later on due to different functionality of the hardware, the fourth software program has been developed to control the vision system on the proposed model.

The first two software programs on the PC (external PC and notebook PC on the proposed model) were developed with VB (using Microsoft Visual Basic), the last two programs were developed in assembly language, this was necessary to handle the incoming activities in real-time.

The first VB program acted as system console on the external PC. This program used SQL calls to support the Microsoft Access database, this was required to create new more commands to maneuver the proposed model, in other words, allowed the user to keep on teaching more capabilities to the robot in the future.

The second VB program resided on the notebook PC inside the proposed model beneath the solar cell panel which was mounted on the back of the robot. This program acted as front end program on the robot to receive commands and

report requested status back to the external PC and communicated with the rest of the 24 Atmel AT89C2051s via RS-485. The program also supported transmission of 2 simultaneous video streams back to the external PC for control information purposes.

Two simple protocols were implemented: the first protocol based on TCPIP platform and the command was embedded in the data portion of the TCPIP protocol. This setup allows the required command/data communication between the programs on both PCs. A series of embedded commands were implemented to allow the program on the notebook PC to keep the incoming commands for itself or to pass them on through the RS-485 interface to all of the Atmel AT89C2051s. The embedded message was a simple one and for one operation only (in case of Atmel AT89C2051, it is for just one particular AT89C2051). Since, the team never built a robot of this magnitude, the design was based on the basic approach at best. This approach allowed implementation of both the individual operation and group (simultaneous) operations. Fig.7. shows typical embedded protocol for AT89C2051

```

: - 1 ST BYTE SOH
: - 2 ND BYTE 1 ST DATA BYTE
:   BIT 0 - ROBOT ID (LSB)
:   BIT 1 - ROBOT ID
:   BIT 2 - ROBOT ID
:   BIT 3 - ROBOT ID
:   BIT 4 - ROBOT ID
:   BIT 5 - ROBOT ID
:   BIT 6 - ROBOT ID (MSB)
:   BIT 7 - ROBOT ID BROADCAST BIT, IGNORE ID CHECKING
: - 3 RD BYTE 2 ND DATA BYTE STATUS
:   BIT 0 - DATA 0 - NO STATUS REQUEST, 1 - STATUS REQUEST
:   BIT 1 - DATA 0 - IMMEDIATE ACTION, 1 - DELAY ACTION
:   BIT 2 - DATA 0 - SWITCH 0 ACTIVE, 1 - ENCODER A ACTIVE
:   BIT 3 - DATA 0 - SWITCH 1 ACTIVE, 1 - ENCODER B ACTIVE
:   BIT 4 - DATA 0 - INDIVIDUAL ACTION, 1 - GROUP ACTION
:   BIT 5 - DATA OPEN
:   BIT 6 - DATA OPEN
:   BIT 7 - DATA OPEN
: - 4 TH BYTE 3 RD DATA BYTE PWM RELATED DATA
:   BIT 0 - DATA PWM DATA (LSB)
:   BIT 1 - DATA
:   BIT 2 - DATA
:   BIT 3 - DATA
:   BIT 4 - DATA PWM DATA (MSB)
:   BIT 5 - DATA OPEN
:   BIT 6 - DATA OPEN
:   BIT 7 - DATA (MSB) CW - 0, CCW - 1
: - 5 TH BYTE 4 TH DATA BYTE PWM RELATED DATA
:   BIT 0 - DATA PWM DATA (LSB)
:   BIT 1 - DATA
:   BIT 2 - DATA
:   BIT 3 - DATA
:   BIT 4 - DATA PWM DATA (MSB)
:   BIT 5 - DATA OPEN
:   BIT 6 - DATA OPEN
:   BIT 7 - DATA (MSB) CW - 0, CCW - 1
:   6 TH BYTE NO. OF PULSES ALLOWED ON ENCODER A
:   7 TH BYTE NO. OF PULSES ALLOWED ON ENCODER B
: - 8 TH BYTE CHECK SUM = 2 ND BYTE + .....7TH BYTE

```

Fig.7. Typical embedded protocol for AT89C2051

For the individual operation, a simple embedded command will be executed, for example, the robot arm can be raised (on one axis) by sending an embedded command to the corresponding AT89C2051 which is responsible for that particular (axis or) motor. For group operation, a series of embedded command (with a holding flag set) will be sent to the respective AT89C2051 in sequential fashion. Once all of the above embedded commands have been transmitted, the final embedded command (with the group action bit set

and the broadcast bit set) will be broadcasted to all of the 24 AT89C2051s and only the AT89C2051(s) which had the delay action bit set (from last command) will move accordingly. Since there is no additional synchronization between the AT89C2051 themselves, the group operation embedded command must be executed with extreme precaution to prevent any possible collision damages to the robot.

The last piece of software was developed to act as video selector for all the 7 cameras and allowed only 2 concurrent video streams to enter the notebook PC. This is necessary due to the limitation of computing power available on the notebook PC. Additional feature on the Microsoft Window allows the above video to flow from the notebook PC to the external PC which acts as a system console of the proposed model.

## VI. TEST METHODOLOGY AND RESULTS

Since the project was carefully analyzed and scaled down to group of small but manageable sub-projects, each sub-projects were carefully tested within the working group and additional tests were made by the involved parties, the final integration of the hardware (on the 2<sup>nd</sup> try) took approximately 4 days and 8+ days after that the proposed model of NASA Mars Rover Spirit was up and running. Since the goal of the project is to emphasis on the concept of Distributed Control, all the tests conducted here are concentrated on the functionality and performance which are related to the Distributed Control concept not the accuracies of the mechanical parts of the robot itself. The following are function and performance tests on the proposed robot, they are:

### A. Functional Test

Various function tests were performed on the proposed model, for examples:

- verification tests on single operation on each individual

Atmel AT89C2051 on the related motors such as: CW, CCW, stop on both limit switches, stop on feedback counts, stop on command from system console, etc.

- verification tests on the series of single command

operations by sending appropriate command to the robot

- one by one to execute consecutive commands.

- verification tests on group operation for some specified

operations which can be done simultaneously, for example, the expansion and retraction of 3 wings, the operation on the 6 driving wheels to maneuver the robot.

### B. Performance Test

After the above functional tests, a series of performance tests was done for the robots as a whole on a testing area of

smooth surface of approximately 30m.x30m. The proposed model of NASA Mars Rover Spirit worked reasonable well on the above performance test.

Comparison of the actual NASA Mars Rover Spirit and the proposed model of the NASA Mars Rover Spirit is in Fig. 8 and Fig. 9 respectively. Fig. 10 is a typical system console screen on the console PC for the proposed Mars Rover Spirit model.



Fig.8. Actual NASA Mars Rover Spirit at JPL Laboratory



Fig.9. Proposed model of the NASA Mars Rover Spirit

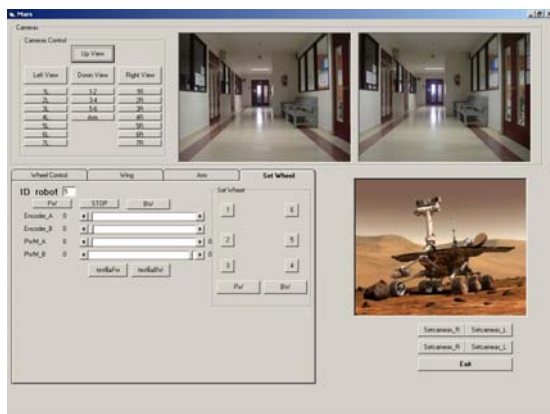


Fig.10. Typical screen on system console PC

## VII. DISCUSSION

In general, the distributed control concept on the proposed model of the NASA Mars Rover Spirit did lay out the actual

ground work of the project. Once the team has applied such concept on the NASA Mars Rover Spirit, separation of sub-projects especially on the control unit (hardware and software) can be done very nicely. The detailed specification of each sub-project was clearly defined and followed. Each sub-project can be done simultaneously on the design, build, test on the same time frame. The final integration of both hardware and software took approximately 2 weeks, the proposed model of the NASA Mars Rover Spirit was up and running. The total budget for this project is well under 2,000 US dollars (the cost of the 2 Notebook PCs is not included).

Since this is the first robot of this type, the team feels that there are still series of redesign and rework need to be done on the next robot which is on the design phase at the moment at SPU. The following are some of the redesign and rework that need to be done on the second proposed model, they are:

- 1) Redesign to make the second proposed model to be at 1/5 size of the actual NASA Mars Rover Spirit. The first one is rather bulky, difficult to move and need a lot of space for any additional research and testing. The aim of the second proposed model is to eliminate any bulky design in the first model.
- 2) FPGA technology is likely to replace the existing network of 24 Atmel AT89C2051s on the first model.
- 3) Notebook PC on the first model will be replaced with a high-end PC-104 or equivalent, this is necessary to allow the second model to withstand any impacts, accidents or allows the robot to move on a rough surface or moves over some obstacles when necessary.
- 4) Computer vision control system will replace the manual focus cameras on the first model.
- 5) The solar cell will be installed that same way as those on the original Mars Rover Spirit including the respective supporting circuits.
- 6) Other modifications if necessary.

Additional researches will be done on implementing Artificial Intelligence capability on the second model. This will allow the new researchers and students in the area of AI, robotics, data communication, image processing, and many other related areas do related researches at the same time at SPU in the very near future.

## VIII. CONCLUSION

The concept of the Distributed Control is working quite well when we properly apply them to the project. This paper is a good demonstration of the above concept, the project ran smoothly and the final integration took only 2 weeks instead of several months when compared with other projects of the same magnitude at SPU. But not all projects can use this concept, it is up to the designers and engineers to review the project thoroughly to make sure that there aren't any flaws left behind. Synchronization is the key issue on the concept of Distributed Control that needs to be

addressed carefully and seriously. Another concerned issue is the bandwidth of the communication link between each control units in the system which can jeopardize the overall system performance later on.

#### REFERENCES

- [1] Distributed Control System on Small-sized RoboCup Jugkree Pala ka Wong Na Adyuthya, Chirit Chiritkuan, Janjai Bhuripanyo, the 2<sup>nd</sup> IEEE International Conference on Cybernatics & Intelligent Systems & Robotics, Automation & Mechatronics. June 7-9, 2006. Bangkok, Thailand.
- [2] Distributed Sensor Fusion for Object Position Estimation in Multi-robot Systems. Ashley W.Stroupe, Martin C.Martin & Tucker Balch The Robotics Institute, Carnegie Mellon University, Pittsburgh, PA.
- [3] Coordination of distributed fuzzy behaviors in mobile robot control, Tunstel, E.; , IEEE International Conference on Volume 5, Issue , 22-25 Oct 1995, vol.5, Page(s):4009 - 4014
- [4] Distributed Robotic Systems, *Hajime Asama*, Journal of Robotics and Mechatronics Vol.8, No.5, pp. 395-399 ,1996
- [5] NASA web site: <http://marsrovers.nasa.gov/home/index.html/>