

Embedded Real-Time Software Architecture for Unmanned Autonomous Helicopters

Won Eui Hong, Jae-Shin Lee, Laxmisha Rai, and Soon Ju Kang

Abstract—The UAV (Unmanned Aerial Vehicle) systems like unmanned autonomous helicopters are used in various missions of flight navigation and used to collect the environmental information of the surroundings. To realize the full functionalities of the UAV, the software part becomes a challenging problem. In this paper embedded real-time software architecture for unmanned autonomous helicopter is proposed that guarantee real-time performance of hard-real time tasks and re-configurability of soft-real time and non-real time tasks. The proposed software architecture has four layers: hardware, execution, service agent and remote user interface layer according to the reactivity level for external events. In addition, the layered separation of concurrent tasks makes different kinds of mission reconfiguration possible in the system. An Unmanned autonomous helicopter system was implemented (Kyosho RC Helicopter) in our lab to test and evaluate the performance of the proposed system.

Index Terms—Unmanned Aerial Vehicle, real-time, concurrent tasks, remote monitoring, sensor data

I. INTRODUCTION

The auto-piloted helicopter as a typical UAV system is self-operated aircraft, used to collect the environmental information of the surroundings, and it decides the tasks

based on its analyzed decisions. To realize the autonomous behavior, the auto-piloted helicopter system consists of complex hardware components and software components and required highly time constrained coordination between these components. Especially, the software part as the abstracted components of hardware consists of three grouped concurrent tasks (hard real-time tasks, soft real-time tasks and non-real-time tasks) and these are always run concurrently. In order to prevent any accidental situation, the deadline of the hard real-time tasks should be guaranteed explicitly and to support the run time flexibility between concurrent tasks (e.g., change flying mode from hovering to go forward), some dynamic reconfiguration for concurrent tasks are also inevitably required. Due to the important applications and technical difficulty, many researches have been tried to solve the auto-pilot helicopter problems[1,2,3,4,5] but majority of these research are considering the auto-pilot issue only and they didn't suggest more flexible software architecture for supporting dynamic reconfiguration of concurrent tasks without loss of any hard deadline of real-time tasks. We focused our research on a hierarchical concurrent task structuring issue without the loss of missing deadline problems. Therefore, this paper proposes a layered software architecture for guarantee the hard real-time constraints and for supporting dynamic reconfiguration simultaneously.

II. RELATED WORKS

The development of autonomous helicopters for real-world applications is a challenging area of research in

Manuscript received October 20, 2005; revised December 3, 2005.
School of EECS, Kyungpook National University (KNU), Korea
E-mail : laxmisha@ee.knu.ac.kr

recent years. The Carnegie Mellon Robotics Institute, developed CMU autonomous helicopter, to build high-accuracy topological maps for NASA geologists [1]. The Georgia Institute of Technology (GIT), designed and developed an UAV avionics system that provides navigational and terrain information [2]. An embedded system for the flight control of an autonomous helicopter has been developed at the Institute for Computer Systems (ICS) at ETH Zurich [3]. In this system, to maintain the body of the helicopter in stable attitude and reduce the misbehavior the PID controller is used. The University of California at Berkeley developed Giotto-based control software by implementing the autopilot system of an autonomous model helicopter. The high performance, in hard real-time embedded systems is achieved using the Giotto approach [4].

III. DESIGN CONSIDERATIONS AND CONCEPTUAL DESIGN

The main task of autonomous flight system is to maintain deadline to avoid any critical conditions. The system must analyze the sensor data and controls the servomotors based on the analysis. The task deadlines must be met within the timeframe to avoid the out-of control situations of the aircraft. The objective of the autonomous flight is not just limited to flight and its control. It has many other various missions like, obtaining pictures or movies using camera, transporting equipments from station to station, gathering weather details etc. These tasks are nonperiodic as well as non-predictable. Execution of these tasks and scheduling of these tasks are also our design considerations. In case of scheduling these tasks, the real-time tasks have to be given higher priority and other tasks with lower priority.

1. Proposed Software Architecture

The overview of the proposed software architecture with guaranteed real-time task execution features is as shown in Fig.1. The proposed software architecture is a layered architecture with four layers as hardware layer, execution layer, service agent layer (SA) and remote user

interface (RUI) layer. The hardware layer composed of sensors, actuators and communication modules and the actual physical module of the autonomous helicopter. Guaranteed real time task execution of the autonomous flight is part of the execution layer.

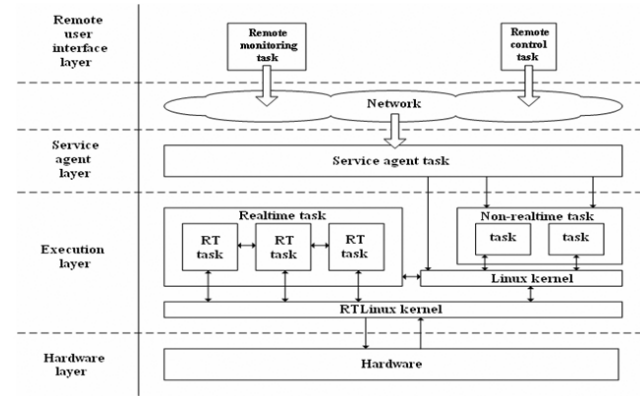


Fig. 1. Overview of proposed software architecture.

The execution layer executes the autonomous flight task, which has the highest priority and also the task which controls the flight operation. The non-real time tasks in the execution layer are executed after the completion of real time tasks. The RUI layer has two functions remote monitoring (RM) and remote control (RC) task and communicates with SA layer using network. The SA task receives the required information from the RM task through the network and decides the state of the helicopter. Using this architecture, we can execute tasks parallel using multiple processors in each layer. Fig.2 shows the hardware system architecture of the proposed UAV helicopter.

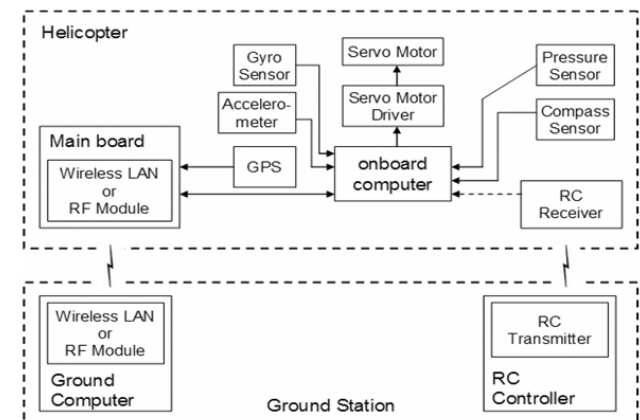


Fig. 2. Hardware system of the proposed UAV helicopter.

IV. DETAILED DESIGNS: SOFTWARE ARCHITECTURE OVERVIEW

The operator first performs the take-off operation, hovering and initial adjustments of the flight using remote controller, manually. Later, the control is switched to the automatic-mode and the helicopter operates automatically and the ground station monitors the state of the helicopter. The monitoring task is executed from the ground station and it provides the information and status about the execution of autonomous flight task of the helicopter.

1. Behavior Analysis

The event sequence diagram is as shown Fig.3.

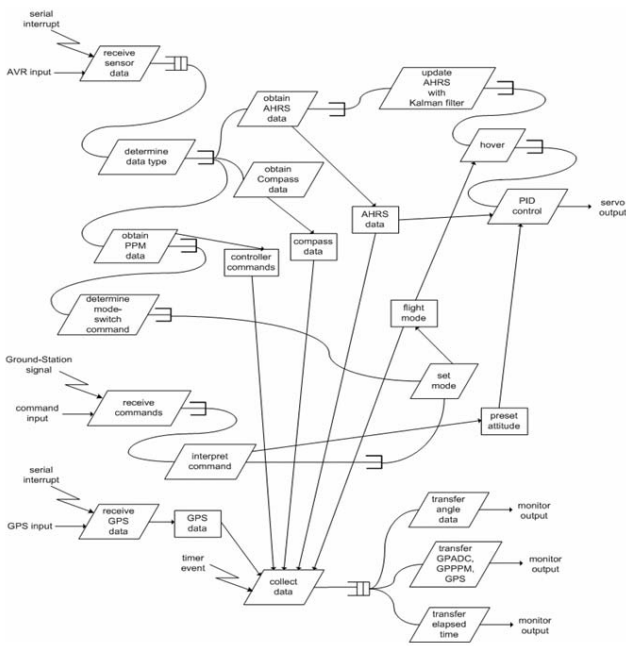


Fig. 3. Event Sequence diagram.

The system gets the quantized IMU (Inertial Measuring Unit) sensor's data, PPM(Pulse Position Modulation) signal compass sensor data from the remote controller using serial port. The program calculates the flight altitude using Kalman filter. Based on the information collected from IMU sensor and compass sensor, the angular velocity, slope information and attitude is calculated in a single task and it is executed periodically. This information is sent to the ground station for monitoring the

flight. The GPS data is also collected and sent to the ground station periodically. The sequence of operations is shown in the Fig.4.

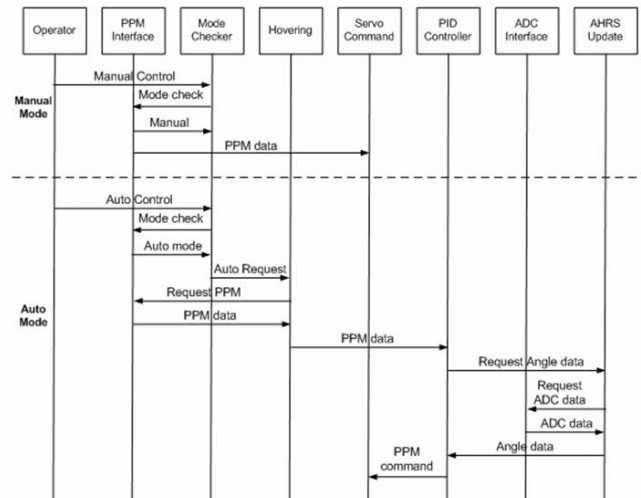


Fig. 4. Sequence diagram.

2. Scheduling Analysis: Onboard and Main Board

According to the proposed hardware configuration as shown in the Fig.2 the scheduling the tasks are based on the tasks running on two processor boards, main board and on-board. On board tasks executed sequentially using round-robin scheduling. First, the onboard receives sensor data from the ADC converter, PPM signal from the remote controller and required data from the main board, later it sends the output to the servo motors. These tasks executed sequentially and repeatedly using fixed time periods. On the other hand, the main board processor executes tasks in multiple priority levels. These tasks are divided as higher priority, intermediate priority and lower priority tasks. The higher priority tasks include tasks to receive the sensor data from IMU, updating AHRIS (Attitude and Heading Reference System) information and obtaining required information from onboard. Based on these data, the task computes the required output data using PID controller. Since, these tasks have highest priority, as they are directly related to flight operation, are not interrupted by any other tasks.

These tasks have fixed running time and only after the execution of these tasks other lower priority asks will run. The intermediate tasks include task to adjust attitude parameter and task to receive GPS data. The Fig.5 shows hovering scheduling diagram.

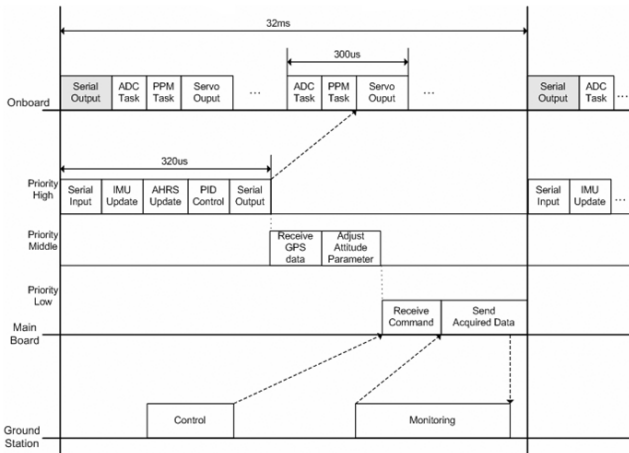


Fig. 5. Hovering Scheduling diagram.

V. IMPLEMENTATION AND PERFORMANCE EVALUATION

1. Implementation

In our research work, we have selected an auto-pilot helicopter as a typical UAV system and named it as KNU Auto-pilot Helicopter. As shown in Fig. 6, the ERGO 50 model of JR PROPO and Concept 60 model of Kyosho type of RC Helicopters are used in this research. It has enough payload capacity to carry, main board, onboard, various sensor and wireless LAN equipments.



Fig. 6. KNU RC Helicopter.

The system is implemented in RTLinux 3.1. The related hardware includes Kyosho RC Helicopter, Transmeta Crusoe compatible x86 processor (Main board), ATM Mega 163 chip (Onboard). The communication protocols used are TCP/IP and Wireless LAN.

2. GUI for Remote Monitoring and Control.

The Fig.7 shows the monitoring and remote control

program output, which is written in Java. The system can observe various data from the sensors using this program. This includes quantized IMU data, transmitted from the onboard to main board, PPM data sent from the receiver to the system and GPS data. The condition of the flight can be observed, based on these data in ground station. The angle information can be analyzed by the operator along with attitude of the current time, which is obtained by the IMU data value. The program displays the change of this value with hour axis in a graph, so that, we can confirm the sensor value collected over a time. The program analyses the PPM signal, which is received from the controller also guides the operator to operate remotely and it displays the information about shifting of manual control to automatic mode of operation.

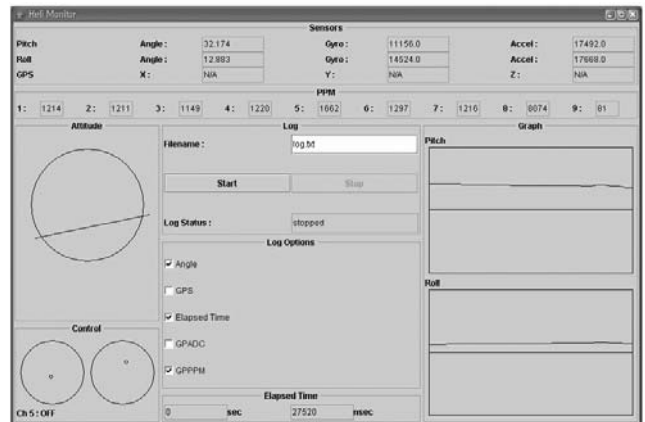


Fig. 7. Remote Monitoring and control program.

The control program also includes the function to store the data collected by the aircraft to a log file. The data is stored from a start time to a specified time of flight operation, by pressing start and stop buttons. If we send the command to system that in order to obtain an execution time, it gets the execution time of flight related task from system, and it indicates that time on the screen. The IMU data, GPS data and PPM data, which are obtained, are verified using sensor data monitoring. Also, a user can confirm the correctness of data transferred from various sensors to unmanned autonomous aircraft. Also user can monitor the state of the remote controller.

VI. CONCLUSIONS

In this paper, software architecture for Unmanned Aerial Vehicle is proposed that guarantees the realtime performance of autonomous flight and the reconfigurability of missions of UAV. The proposed system architecture is divided into hardware layer, execution layer, service agent layer and remote user interface layer. With this architecture; we can apply this to the actual unmanned helicopter and can confirm its exact operation, based on performance tests and results. Also, this architecture can be applied to various unmanned control systems.

VII. ACKNOWLEDGMENTS

This work was supported by grant No. R01-2003- 000-10252-0 from the Basic Research Program of Korea Science and Engineering Foundation.

REFERENCES

- [1] Ryan Miller , Omead Amidi, and Mark Delouis, Arctic Test Flights of the CMU Autonomous Helicopter, Carnegie Mellon Robotics Institute, <http://www.ri.cmu.edu/project/chopper>.
- [2] Joerg S Dittrich and Eric N Johnson. A Multi-Sensor Navigation System for an Autonomous Helicopter, AIAA Digital Avionics Conference and No. 377, Irvine, CA and Oct. 2002.
- [3] Markus Kottmann Software for Model Helicopter Flight Control, Technical Report 316, Institute of Computer Systems and ETH Z, 1999.
- [4] Christoph M. Kirsch, Marco A.A. Sanvido, Thomas A. Henzinger, A Giotto-based Autonomous Flying Helicopter System, University of California at Berkeley, Berkeley, USA.
- [5] Myungsoo Jun and Stergios I Roumeliotis and Gaurav S Sukhatme, State Estimation of an Autonomous Helicopter Using Kalman Filtering. Proc. 1999 IEEE/RSJ International Conference on Robots and Systems (IROS 99).
- [6] Eric N Johnson and Paul A Debitetto, Modeling and Simulation for Small Autonomous Helicopter Development, American Institute of Aeronautics and Astronautics, 1997.
- [7] M Morin, S Nadjm-Tehrani, P Osterling, and E Sandewall, Real- Time Hierarchical Control, IEEE Software and Vol. 9, pp51 -57, Sept. 1992.
- [8] Hanssan Gomaa, Designing Concurrent Distributed and Real-Time Applications with UML and Addison Wesley, 2000.
- [9] Paul G Fahlstrom and Thomas J Gleason, Introduction to UAV Systems, UAV Systems INC, 1998.
- [10] R/C Helicopters, <http://www.easyrc.com/helicopters/>



Won Eui Hong He received B.S and M.S degrees from Department of Electronics Engineering, Kyungbook National University, Daegu, Korea, in 2003 and 2005 respectively. He is currently working in Agency for Defense Development (ADD), Korea. His research interests include Home Networks, Real-time Systems and Software Architecture for Unmanned Aerial Vehicles.

Korea Atomic Energy Research Institute (KAERI) as a member of research staff and a head of computing and information research department. Since September 1996, he has been a faculty member of School of Electronic and Electrical Engineering of Kyungpook National University. His current research interests include real-time system, home network, distributed object technology and software engineering. Dr. Kang is a member of IEEE computer society, KISS, KIEE, KITE.



Jae-Shin Lee He received B.S degree from Department of Electronics Engineering, Kyungbook National University, Daegu, Korea, in 2004. He is currently pursuing the M.S degree in the Department of Electronics Engineering at Kyungbook National University. His interests include CAN based system and Real-time OS for Unmanned Aerial Vehicles.



Laxmisha Rai He is working towards the Ph.D degree in the school of EECS at Kyungpook National University, Korea. He received his B.E. degree in Computer Engineering from KVG College of Engineering, Mangalore University and M.Tech, degree in Computer Science and Engineering from Manipal Institute of Technology, MAHE, Deemed University, India in 1998 and 2001 respectively. His research interests include Software Engineering, Real-Time Operating Systems, Real-Time Embedded Software and Autonomous Mobile Robots. He is a Student Member of IEEE.



Soon-Ju Kang He received the B. S. degree in electronics from Kyungpook National University, Daegu, Korea, in 1982, and the M. S. and Ph. D. degrees in computer science from Korea Advanced Institute of Science and Technology (KAIST), Taejon, Korea in 1985 and 1995, respectively. From 1985 to August 1996, he worked at