Control Architecture for Stable Gait of a Six-legged Subsea Robot CR200

Banghyun Kim, Hyungwon Shim, Seong-yeol Yoo, Gyeong-mok Lee, Bong-Huan Jun, and Pan-Mook Lee
Ocean System Engineering Research Division
MOERI, KIOST
Daejeon, Republic of Korea

***Abstract*—This paper presents a control architecture for 200m-class subsea walking robot, named CR200, which can work in strong tidal current and high turbidity environment by crawling sea floor to survey shipwreck and scientific research. The control architecture consists of deliberative layer, execution layer, and hardware abstract layer. Deliberative layer is in charge of the high level control of the CR200. Deliberative layer only gets needed information from execution layer, and puts processing result to execution layer. Deliberative layer does not consider detail process because execution layer includes most essential process for robot control. Hardware abstract layer allows other layers to perform device-independent, and it includes communication module, sensor interface module, and motor interface module. The control architecture has been implemented to a program of operating software. The program runs on Linux operating system, and it supports soft real-time characteristic without real-time operating system. The control frequency of the CR200 is 100 Hz, and standard deviation of software timer occurrence error was 10.375 us. We conducted first sea trial of the CR200 at KIOST's South Sea Research Institute in July 2013, and the CR200 performed well during its first die.**

Keywords— control architecture; subsea robot; walking robot; CR200; Crabster

I. INTRODUCTION

Developed at the Korean Institute of Ocean Science and Technology (KIOST), the CR200 is an alternative to propeller-driven remotely-operated vehicles (ROVs) and autonomous underwater vehicles (AUVs), which are ill-equipped to deal with strong tidal currents in shallow seas. Robotic underwater vehicles play an important role in marine research, environmental operations, deep water exploration, and search-and-rescue missions. Most ROVs and AUVs rely on propellers, which allow them to maneuver swiftly and dive to great depths. But there are certain places these machines can't easily access because of strong currents. To overcome this hurdle, we sought inspiration not from fish but from legged sea creatures like crabs and lobsters. The result was CR200 [1].

In total the CR200 is 2.42 m long, 2.45 m wide, and 1.3 meter high. The height rises until 1.8 meter when it stands up. The weight of the robot is about 640 kg. Operation depth is up to 200 meter and maximum walking speed is 0.5 m/s. The CR200 is equipped with an ADCP (Acoustic Doppler Current Profiler), an AHRS (Attitude and Heading Reference System), a CTD (Conductivity Temperature Depth), an USBL (Ultra Short Base Line), an altimeter, a tool sled, LED lights, a Gigabit Ethernet hub, seven Ethernet-to-CAN converters, an Ethernet-to-serial converters, a single-beam scanning sonar, a multi-beam acoustic camera, seven analog optical cameras, a HD network cameras, and two video encoders. Six legs have 30 motors, 28 incremental encoders, 24 absolute encoders, and 6 loadcells.

This paper introduces a control architecture for stable gait of the CR200. The control architecture consists of deliberative layer, execution layer, and hardware abstract layer. They includes only mandatory modules to control the CR200 except user interface and sensor monitoring. The control architecture supports 100 Hz soft real-time control required for stable gait of the CR200 using software timer without real-time operating system (RTOS). Generated control messages are transferred from remote control unit to motors through Ethernet network, optical fiber network, and CAN bus.

We conducted on-lend tests and first sea trial at KIOST's South Sea Research Institute (SSRI) in July 2013 as shown in Fig. 1. The CR200 walked and changed posture well on the seabed 5 to 7 meters depth. We also tested LED lights, optical cameras, scanning sonar, acoustic camera, and tool sled.

Fig. 1. Recovery after subsea walking test

This work was supported by the Ministry of Oceans and Fisheries (MOF) of Korea for "Development of a Multi-legged Walking Flying Subsea Robot".

The remainder of this paper is organized as follows. Section II shows communication system for remote control of legs, and Section III describes the operating software of the CR200. Section IV introduces our control architecture for stable gait of the CR200, and the last section provides concluding remarks and the future of our work.

II. COMMUNICATION SYSTEM FOR LEGS

The structure of CR200 system is similar to traditional ROV system which consists of remote control unit, depressor, and vehicle. The CR200 acts like other ROVs by remote commands from remote control unit. The CR200 is connected to the remote control unit through 300 meter cable which includes six optical fiber lines with 1,310 nm diameter and 150 VDC electric lines. The fiber optic Gigabit media converter with 1,000 Mbps bandwidth interfaces optical fiber network with Ethernet network. The main network of CR200 system is Gigabit Ethernet with 1,000 Mbps bandwidth. All devices with Ethernet interface are interconnected by Gigabit Ethernet switch. Other devices without Ethernet interface are connected to Ethernet network using Ethernet convertor [2].

Six legs have 30 motors, 28 incremental encoders, 24 absolute encoders, and 6 loadcells. They within each leg are interconnected using CAN bus with 1 Mbps bandwidth. Each CAN bus is connected to Ethernet network using Ethernet-to-CAN converter and AHRS with RS-232 interface is connected to Ethernet network using Ethernet-to-Serial converter. Communication system for leg control is as shown in Fig. 2. "L1M1" means motor driver of first joint in front left leg, "R3M4" means motor driver of fourth joint in rear right leg, and so on. "LC" means loadcell for force sensing. Like a crab or a lobster, the CR200's two front legs are equipped with manipulators that can grasp objects that can be stored in a frontal compartment, so front leg has three more motors compared with other legs.

User control instructions of the CR200 from pilot computer and copilot computer using joysticks are sent to agent computer. Motor control messages are generated in agent computer by control instructions, and they are sent to motor drivers through Ethernet network, optical fiber network, and CAN bus.

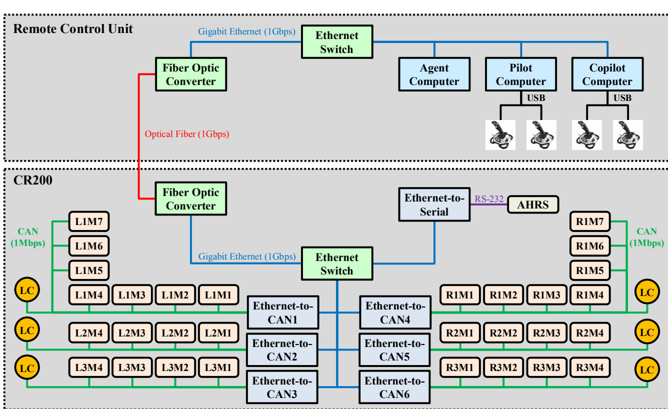

Fig. 2. Communication system for leg control

III. OPERATING SOFTWARE

Operators remotely controls the CR200 using operating software such as agent program, pilot program, 3D viewer program, dynamics simulation program, navigation program, USBL program, video program, scanning sonar program, acoustic camera program, and current monitoring program as shown in Fig. 3. Agent program, that is main software, interfaces remote control unit with the CR200. Pilot program provides user interface for remote control of the robot, and 3D viewer program displays 3D pose of the robot in real time. Navigation program displays position of robot and ship on map, and USBL program interfaces USBL transducer with remote control unit. Video program shows surrounding using video image of optical cameras while scanning sonar program and acoustic camera program show surrounding using sonar image of sonar cameras. Current monitoring program shows current direction and speed around the robot, and dynamics simulation program can test a posture or gait algorithm before it applies the CR200 [3].

We have developed agent program, pilot program, 3D viewer program, and dynamics simulation program. Navigation program is HYPACK software which provides tools to design a survey, collect data, apply corrections to soundings, remove outliers, plot field sheet, export data to CAD, compute volume quantities, generate contours, create side scan mosaics, and create electronic charts. Video program is Axis Camera Station software which is a digital monitoring and recording system for up to 50 cameras when Axis camera system is used. The CR200 system has four Axis network cameras and seven analog cameras connected two Axis video Encoders. Scanning sonar program, acoustic camera program, and USBL program are provided by device manufacturer. Current monitoring program is BBtalk software and winADCP software for recording and monitoring of ADCP data.

Operating programs run on eight computers in remote control unit. Computers consist of agent computer, navigation computer, video computer, pilot computer, copilot computer, sonar computer, mission computer, and USBL computer. Agent program runs on agent computer, pilot program and 3D viewer program run on pilot computer and copilot computer

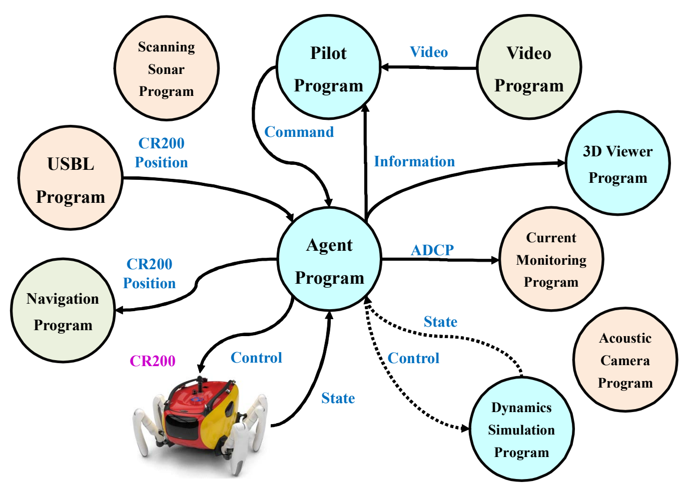

Fig. 3. Organization of operating software for CR200

having dual monitors, dynamics simulation program runs on pilot computer, navigation program runs on navigation computer, video program runs on video computer, scanning sonar program and acoustic camera program run on sonar computer, current monitoring program runs on mission computer, and USBL programs runs on USBL computer. Agent program runs in Linux operating system while other programs run in Microsoft Windows 7 operating system.

Agent program processes received state data from the CR200 and operator's control command from pilot program, makes control message, and sends it to the CR200. In simulation mode, agent program sends the control message to dynamics simulation program, and receives simulation results such as simulated time, joint angle of legs, roll, pitch, yaw, and position from simulation program. Received simulated time increase system time of agent program. Agent program sends various information about the CR200 system to other programs. Control architecture for remote control of the CR200 is implemented in agent program.

IV. Control Architecture

Control architecture for the CR200 consists of deliberative layer, execution layer, and hardware abstract layer as shown in Fig. 4, and they have been implemented in agent program. Deliberative layer is in charge of the high level control of the CR200 during posing and walking. Deliberative layer does not consider detail process because execution layer includes most process of agent program. However, execution layer does not include hardware interface, but manages their data by virtual communication buffer provided by hardware abstract layer which allows execution layer to perform device-independent.

Three layers consists of 15 modules. Gait controller module includes various posture algorithms and gait algorithms. Agent main management module initializes agent program, and interfaces stream output. Multiple process module manages threads including main control loop process, and logical sensor module provides shared data storage to other modules. Shell interface module provides command line interface (CLI) to use agent program, and log module writes state of the CR200 to file. Leg test module includes test functions to move joint of legs, and utility library module provides useful functions such as time management and bit operation. Inter-program communication module encodes or decodes a communication message between other programs using designed protocol.

Virtual communication module provides hardware abstraction of serial and socket communication to upper layer using given name. Sensor interface module decodes a received message from sensor, and control interface module encodes a control message for motor, relay switch, light, and tool sled. When multiple serial module and multiple socket module receive a massage, it is immediately stored into multiple circular queue without any message processing. Decode and process of all received message is periodically performed by main control loop in multiple process module.

Legs control is supported by pilot program and agent program as shown in Fig. 5. Whenever operator moves joystick, leg control instruction is generated and pilot program sends it to agent program in form of shell instruction. Shell interface module decodes the shell instruction to gait or arm control instruction code and stores it to logical sensor. Leg controller module computes joint angle references by gait and arm control command, and stores them to logical sensor. Motor interface module, which is one of control interface module, converts joint angle references to motor control messages, and sends them to 30 motors in six legs.

Main control loop thread is executed repeatedly without RTOS while agent program runs. Main control loop consists of received data process, leg control process, information transmission for other programs, logging, and wait for synchronization. Received data process gets and processes all data stored in multiple circular queue sequentially. Leg control process calculates joint angle references by user control instructions and sends motor control messages to 30 motor drivers. Information transmission for other programs sends periodic information to other programs by given frequency, and wait for synchronization sleeps until next timer occurrence. Wait for synchronization was implemented using system time and *usleep()* function supported by Linux operating system. One turn of main loop process should be finished within 10 ms

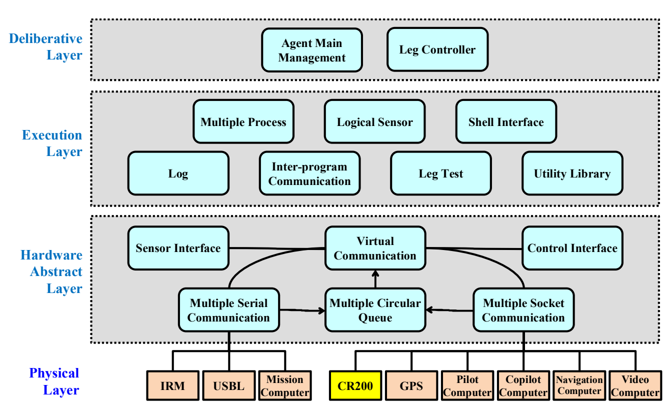

Fig. 4. Control architecture

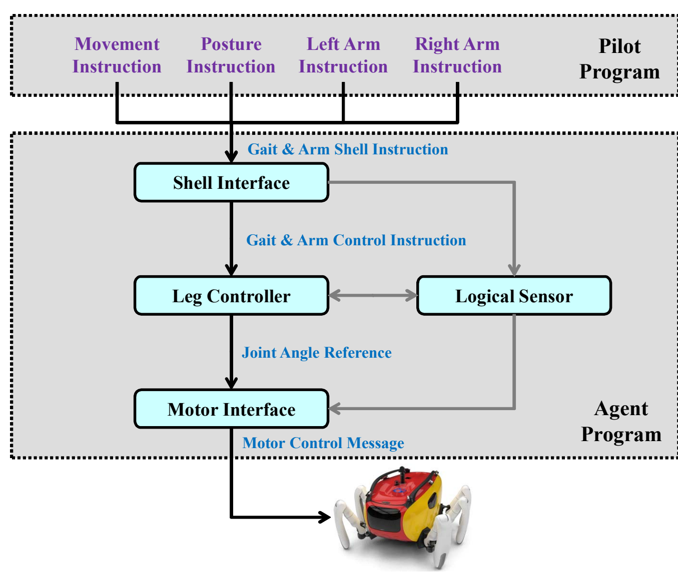

Fig. 5. Legs control flow

for 100 Hz control frequency. A study using software timer [4] show that the real-time management approach is enough to control the CR200.

We measured software timer errors of agent program for two hours when control frequency is 100 Hz. Agent program ran on agent computer which has 2 GHz Intel Quad Core i7 CPU, 8 GB main memory, and Gigabit Ethernet interface. By experimental results, average of timer occurrence interval was exact 10 ms. Also, timing occurrence error is less than 171 us, and its standard deviation is 10.375 us. Although these results means that software timer provides soft real-time control, our approach without RTOS is enough to control the CR200 by 100 Hz frequency because underwater robot such as CR200 moves slowly.

Process of leg controller is shown in Fig. 6. Gait algorithms can accept new control instructions only if leg state is "Ready". Current algorithms have a limitations which is that other action is impossible until a cycle of running action ends, so we defined "Ready" state. When the leg state is "Ready", leg controller fetches recent control instructions stored in logical sensor. Instructions are generated by joysticks of pilot computer and copilot computer, and consists of left arm instruction, right arm instruction, movement instruction, and posture instruction as shown in Fig. 7. We defined instruction codes by byte unit for bit operation.

Current gait algorithms can be selected in exclusive gait and composite gait. Exclusive gait does not support simultaneous movement and posture, but composite gait support it. In composite gait, arm control is also possible simultaneously. In exclusive gait, arm control is possible only if movement control and posture control are "Stop". Gait algorithms uses four legs or six legs. Arm control is always "Stop" in six legs gait.

Gait algorithm computes next angle of 24 joints by current gait control (GC), and arm controller computes next angle of 14 joints in two front legs by current arm control (AC). Motor interface module converts computed reference angles to motor control messages, and sends them to 30 motor drivers. This process repeats by 100 Hz.

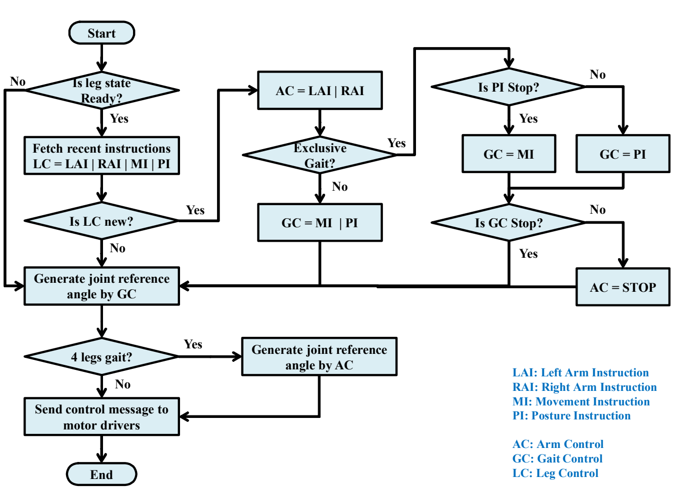

Fig. 6. Process of leg controller

V. Conclusion

We have implemented the control architecture presented in this paper, and finished unit test, integration test, on-land test, and sea trial. The CR200 performed well during its first sea trial in real-world conditions. There were some minor problems but we were satisfied the results. Now improvement works of the CR200 system is in progress by the results of tests. Next year, we will conduct sea trials in strong tide sea and more deep sea.

References

[1] J. Falconer, "Huge Six-Legged Rogot Crabster Goes Swimming," IEEE Spectrum, July 2013.

[2] B. Kim, S.W. Park, P.M. Lee, and B.H. Jun, "Design of Communication and Video System for a Multi-legged Subsea Robot," Oceans'11 Santander, 2011.

[3] B. Kim, H. Shim, S.W. Yoo, B.H. Jun, S.W. Park, and P.M. Lee, "Operating Software for a Milti-legged Subsea Robot CR200," Oceans'13 Bergen, 2013.

[4] B. Kim, B.H. Jun, J.Y. Park, and P.M. Lee, "Real-time Process without RTOS for the ISiMI100 Autonomous Underwater Vehicle," Oceans'12 Yeosu, 2012.

Left Arm Instructions	
ARML_STOP	**0x0000**
ARML_J1PLUS	**0x1000**
ARML_J1MINUS	**0x2000**
ARML_J2PLUS	**0x3000**
ARML_J2MINUS	**0x4000**
ARML_J3PLUS	**0x5000**
ARML_J3MINUS	**0x6000**
ARML_J4PLUS	**0x7000**
ARML_J4MINUS	**0x8000**
ARML_J5PLUS	**0x9000**
ARML_J5MINUS	**0xa000**
ARML_J6PLUS	**0xb000**
ARML_J6MINUS	**0xc000**
ARML_GRIPPER_OPEN	**0xd000**
ARML_GRIPPER_CLOSE	**0xe000**

Right Arm Instructions	
ARMR_STOP	**0x0000**
ARMR_J1PLUS	**0x0100**
ARMR_J1MINUS	**0x0200**
ARMR_J2PLUS	**0x0300**
ARMR_J2MINUS	**0x0400**
ARMR_J3PLUS	**0x0500**
ARMR_J3MINUS	**0x0600**
ARMR_J4PLUS	**0x0700**
ARMR_J4MINUS	**0x0800**
ARMR_J5PLUS	**0x0900**
ARMR_J5MINUS	**0x0a00**
ARMR_J6PLUS	**0x0b00**
ARMR_J6MINUS	**0x0c00**
ARMR_GRIPPER_OPEN	**0x0d00**
ARMR_GRIPPER_CLOSE	**0x0e00**

Movement Instructions	
MOVE_STOP	**0x0000**
MOVE_FORWARD	**0x0010**
MOVE_FORWARD_R	**0x0020**
MOVE_SHIFT_RIGHT	**0x0030**
MOVE_BACKWARD_R	**0x0040**
MOVE_BACKWARD	**0x0050**
MOVE_BACKWARD_L	**0x0060**
MOVE_SHIFT_LEFT	**0x0070**
MOVE_FORWARD_L	**0x0080**
MOVE_ROTATE_LEFT	**0x0090**
MOVE_ROTATE_RIGHT	**0x00a0**

Posture Instructions	
POSE_STOP	**0x0000**
POSE_READY	**0x0001**
POSE_ROLL_INC	**0x0002**
POSE_ROLL_DEC	**0x0003**
POSE_PITCH_INC	**0x0004**
POSE_PITCH_DEC	**0x0005**
POSE_YAW_INC	**0x0006**
POSE_YAW_DEC	**0x0007**
POSE_SURGE_INC	**0x0008**
POSE_SURGE_DEC	**0x0009**
POSE_SWAY_INC	**0x000a**
POSE_SWAY_DEC	**0x000b**
POSE_HEAVE_INC	**0x000c**
POSE_HEAVE_DEC	**0x000d**

Fig. 7. Definition of leg control instructions