Distributed Self-reconfiguration Control of an M-TRAN System

Kurokawa H, Tomita K, Kamimura A, Kokaji S
National Institute of Advanced Industrial Science and Technology

Murata S
Tokyo Institute of Technology
Modular Robot
  - Modular robot as a DARS
  - Problem of self-reconfiguration

M-TRAN
  - Hardware
  - Software

Experiment

Conclusion
M-TRANs

I  II  III
1998  2002  2005
66 mm  60 mm  65 mm
440 g  400 g  420 g

I  Basic Experiments (centralized)
II  Locomotion by Distributed Control
III  Self-reconfiguration by Distributed Control
Self-reconfiguration (metamorphosis)

2-dimensional system (Fracta 1992)
Self-reconfiguration by distributed autonomous method was successful
(Decentralized, asynchronous, neighbor-to-neighbor communication)

3-dimensional system
Small scale self-reconfiguration by centralized or globally synchronous controller

3-D module
Key factors for self-reconfiguration experiments

- Basic design of a module is important. There is no optimal design. M-TRAN is a candidate.

- Hardware design & performance:
  - speed
  - power consumption
  - reliability

- Cost is important for mass production.
  - 100 ATRON
  - 50 M-TRAN III
  - ...
  - 1000 ???

ATRON (U.S.Denmark)
Experiments in the past

M-TRAN II
10 modules
(20 modules produced)
# of connection changes
1 or 2 times/module
8 min.

ATRON (USD)
35 modules
(100 in total)
# of connection changes
<10 times (2 module)

CONRO (USC)
one disconnection
one connection
Distributed Metamorphosis Control

• Module design:
  Symmetric omnipotent module: too heavy
  Fewer DOF, fewer symmetry:
    Complicated motion for self-reconfiguration

→ Simple design (M-TRAN)
→ Regular structure and repetitive self-reconfiguration
  (suitable for parallel control)

• Hardware performance:
  speed, power consumption, reliability
  especially of connection mechanism

• Small production: cost, ...
M-TRAN module (lattice-oriented design)

Rounded-cubic block
Connection surface

180° rotation

Link
Connection

180° rotation

Module
Neighbor module

Key idea

Positioned in cubic lattice by angle=0, ±90°
Stackable in a cubic lattice
Large surface for connection
Avoid collision (parallel axes)
Distributed Metamorphosis Control

• Module design:

• Hardware performance:
  speed, power consumption, reliability
  especially of connection mechanism

→ New mechanism for M-TRAN (Fast, low power consuming, reliable)

• Small production: cost, ...
New M-TRAN Hardware

Module

Motion DOF 2
Connection: male 3, female 3
CPU 1 main / 3 slave
10 IR proximity sensor
Gravity sensor
Global communication by CAN bus
Bluetooth modem
Battery in each module
Mass production:
50 M-TRAN III modules were produced for EXPO 2005
Target procedures for experiments

- Regular structure by meta-modules and repetitive self-reconfiguration

Tomita (2000)
Ostergaard (DARS 2004)
Yoshida (2001)
Zack (2002)
Lund (2003)
Kurokawa (2004)
Kurokawa (2005)
Distributed self-reconfiguration Control

• Advanced module design & self-reconfiguration design

• Improved hardware performance:
  • 50 modules

Research Objective

Experimental verification of

• Decentralized and asynchronous parallel control

• Self-reconfiguration by large number of modules (>20)
Software development

- Onboard controllers
  Master CPU + 3 slave CPUs

- M-TRAN simulator
  - Design of self-reconfiguration procedure (multi-thread, step synchronous)
  - Kinematics & Dynamics Simulation (Vortex & ODE)
Software

M-TRAN simulator
• Step synchronization
• Script conversion

Centralized control by Host PC (M-TRAN I)

Global event synchronization (M-TRAN II)

past

Emulator for distributed controller

Onboard Controller

Asynchronous decentralized control (M-TRAN III)
Onboard Controller System

Host PC

Network bus

(1) Shared memory  (2) Remote control  (3) Message for synchronization
Parallel Controller System

m, 1, 90, 90

ID=1 ID=2

move motors of id=1 to $90^\circ, 90^\circ$

Network bus

Virtual CPU

ID=1

Module

Command Interpreter

Program

Memory

Peripherals

Motors

Connectors

Sensors

Virtual CPU

ID=2

Module

Command interpreter

Program

Memory

Peripherals

(1) Shared memory  (2) Remote control  (3) Message for synchronization
Parallel Controller System

- **Remote, m, 2, 90,90**

Network bus

- **m, 2, 90,90**

- **move motors of id=2 to 90°,90°**

- **Virtual CPU**
  - ID=1
  - ID=2

- **Module**
  - Command Interpreter
  - Program
  - Memory
  - Peripherals
    - Motors
    - Connectors
    - Sensors

- **Bluetoooth**

- **(1) Shared memory**
- **(2) Remote control**
- **(3) Message for synchronization**

- **Host PC**
Parallel Controller System

(1) Shared memory
(2) Remote control
(3) Message for synchronization
Parallel Controller System

Remote 2,A,=,2

Network bus

Host PC

Bluetooth

Command Interpreter

Program

Memory

Peripheral Motors Connectors Sensors

Virtual CPU

ID=1 Module

ID=2 Module

(1) Shared memory

(2) Remote control

(3) Message for synchronization
Parallel Controller System

- Single master control (Remote control)
- Parallel & locally synchronous control (Shared memory)
Program development system

M-TRAN simulator

Simulation Script
while (1) //loop
{
  mov(1, 1, 90, -90)
  mov(2, 0, 90, -90)
}
endwhile

Auto conversion

Machine Program (Single master)
L0: cnr 2, 2, 0
    mvr 1, -90, 90
    mvr 2, 90, -90
    cnr 2, 2, 1
    cnr 1, 2, 0
    mvr 1, 90, -90
    mvr 2, -90, 90
    cnr 1, 2, 0
    jpr L0

Machine Program (parallel)
L0: switch flag, L0, L1, L2, L3
L1: load flag, 0
    con 2, 0
    remote, next, load, flag, 3
    mov -90, 90
    jpr L0
L2: load flag, 0
    con 2, 1
    remote, next, load, flag, 1
    jpr L0
L3: load flag, 0
    mov, 90, -90
    remote, next, load, flag, 2
    jpr L0

L0: switch flag, L0, L1, L2, L3
L1: load flag, 0
    con 2, 0
    remote, next, load, flag, 3
    mov -90, 90
    jpr L0
L2: load flag, 0
    con 2, 1
    remote, next, load, flag, 1
    jpr L0
L3: load flag, 0
    mov, 90, -90
    remote, next, load, flag, 2
    jpr L0

example of parallel control
Emulation of parallel processing

Emulator = M-TRAN Simulator (Kinematics & Dynamics) + multi-Controller Emulation

Source code compatible to the onboard controller

Virtual CPU
Command Interpreter & Memory

Slave CPU control
Link motor connector etc.

Multi-Controller Emulation

Network Bus
Experiments

- Locomotion

- Self-reconfiguration
  - Centralized & synchronous
    (Single master)
  - Decentralized & asynchronous
    (Parallel)
Experiment (single master)

Arbitrary 4 module
Master voting
Identification of configuration & role
Locomotion (clock sync.)
Self-reconfiguration
Single master

Demonstrated in EXPO 2005
Parallel distributed control

Algorithm

- Two modules cooperate (meta-module, cluster)

- Communication with 4 neighbors

translation by self-reconfiguration
Parallel distributed control

# of modules: 4, 8, 12, 20

The same program for all the modules (code size 660 Byte)

Local synchronization
Parallel control

code size ≈ 1.5 KB
using long distance communication

Hardware problems
mechanical: alignment error → connection fails
communication: unreliable electric contact, ...

Improvement
retrial
reconnection
Parallel control (improved a little)

Connection retrial

Neighbor to neighbor communication
Connection retrial

code size ≈ 2 KB
Problems (1)

2 SW for CAN bus lines
1 SW for a line of connection detection

Switches to avoid unexpected short circuit are unreliable
Problem (2)

- Misalignment between surfaces for connection
- Bus traffic jam

Connection failure
Critical message

- message & protocol design is important

Every step is critical
(deterministic procedure)

Most connections are critical for communication

→ redundancy
→ nondeterministic

Redundant connection
Process can be nondeterministic
Future works

1. Larger structure (> 20 modules)

2. Autonomous self-reconfiguration by sensor information

3. Automatic separation of faulty module
Future works (detail)

- Neighbor-to-neighbor communication by IR devices
  Reliable, scalable, slow (100 bps)

- Controller language
  Assembler language → High level

- Sensing and decision making
Experiment (single master)

Single Master Playback
Verification of hardware performance

Autonomous path following
Summary

- Development of a new Hardware (M-TRAN III) and software

- Distributed controller for centralized & decentralized self-reconfiguration control

- Simple parallel self-reconfiguration was verified by experiments