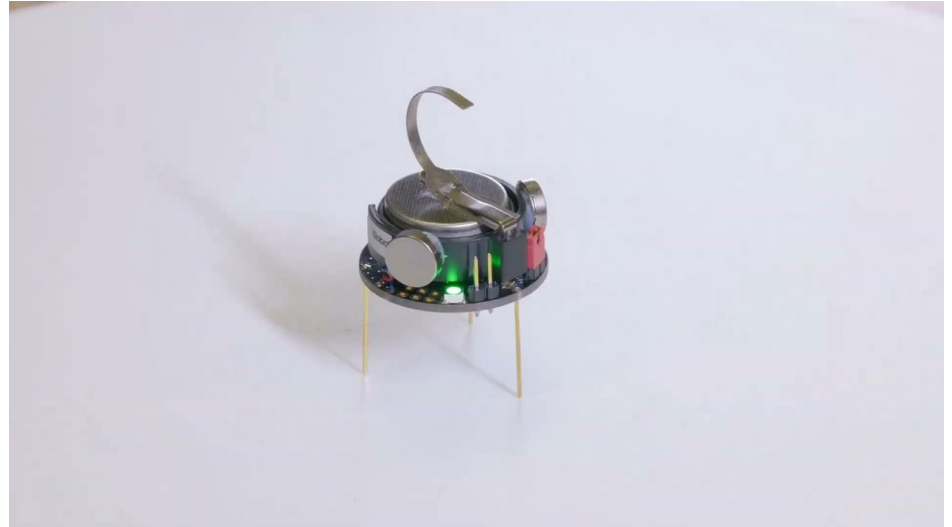


Nadzoru 2 and Kilobot

Genki Miyauchi

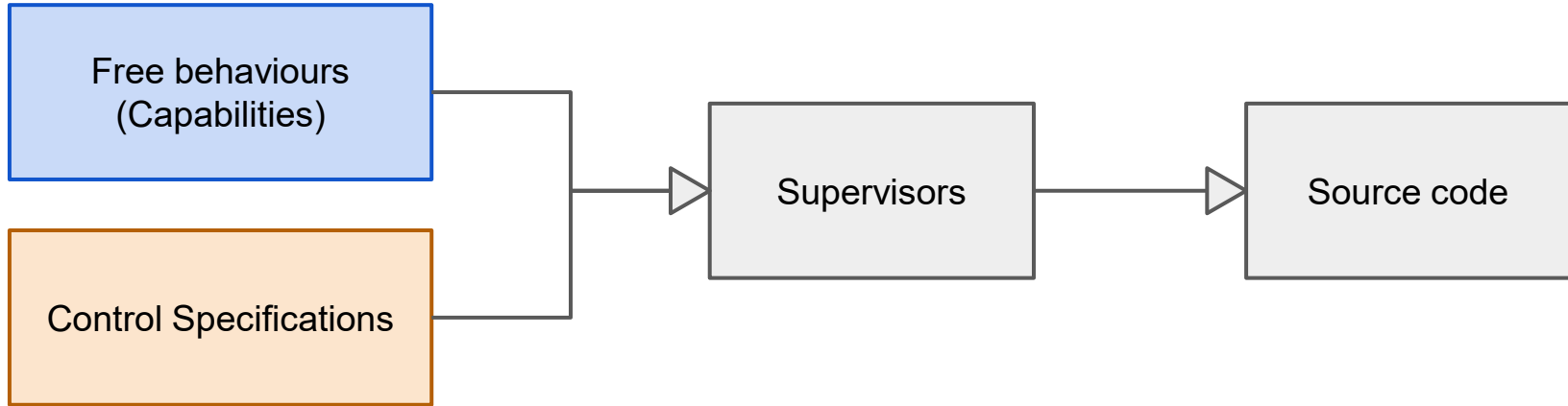
Kilobot

- **Body size:** 3.3 cm
- **Lithium-ion battery** (3-12 hours)
- **2 vibration motors** are used to move forward and rotate
- **Infrared transmitter and receiver** are used for (i) uploading new programs to the Kilobot and (ii) achieving communication between Kilobots (10 cm)



Nadzoru 2

Supervisory Control Theory (SCT)



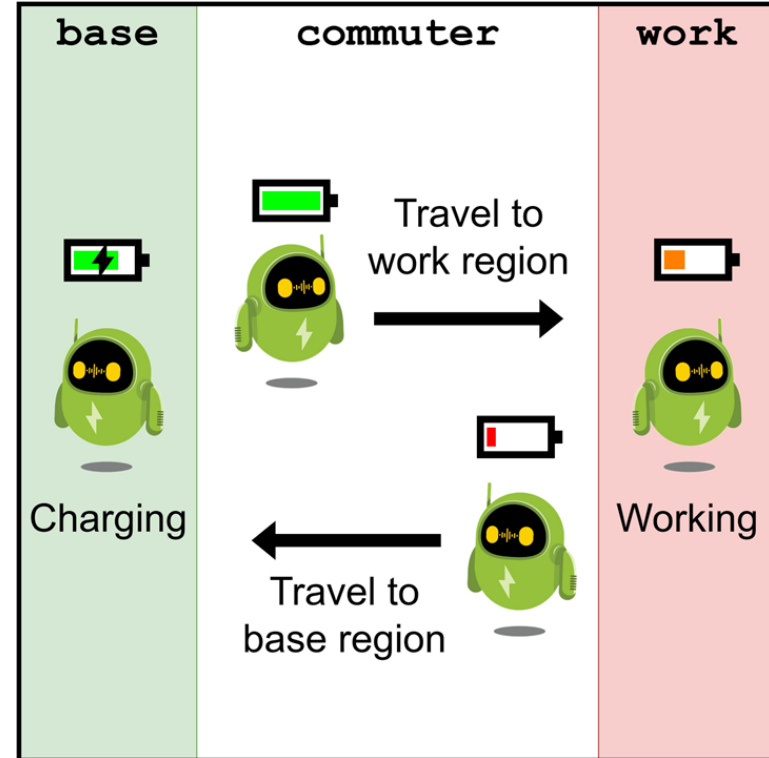
Example Scenario

Design a robot that:

- carries out some **work**
- returns to the base to **recharge**

Robot's capability:

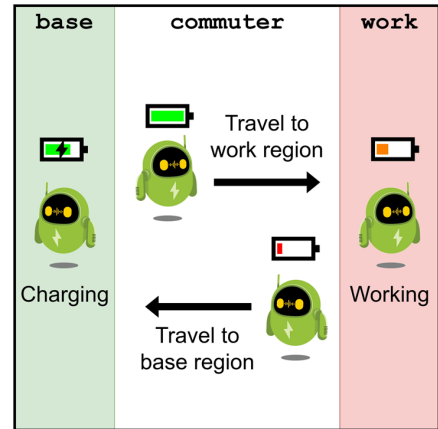
- **Move** between the base and work regions
- **Charge/Work** in the base and work regions
- **Detect** whether it is inside the base or work region



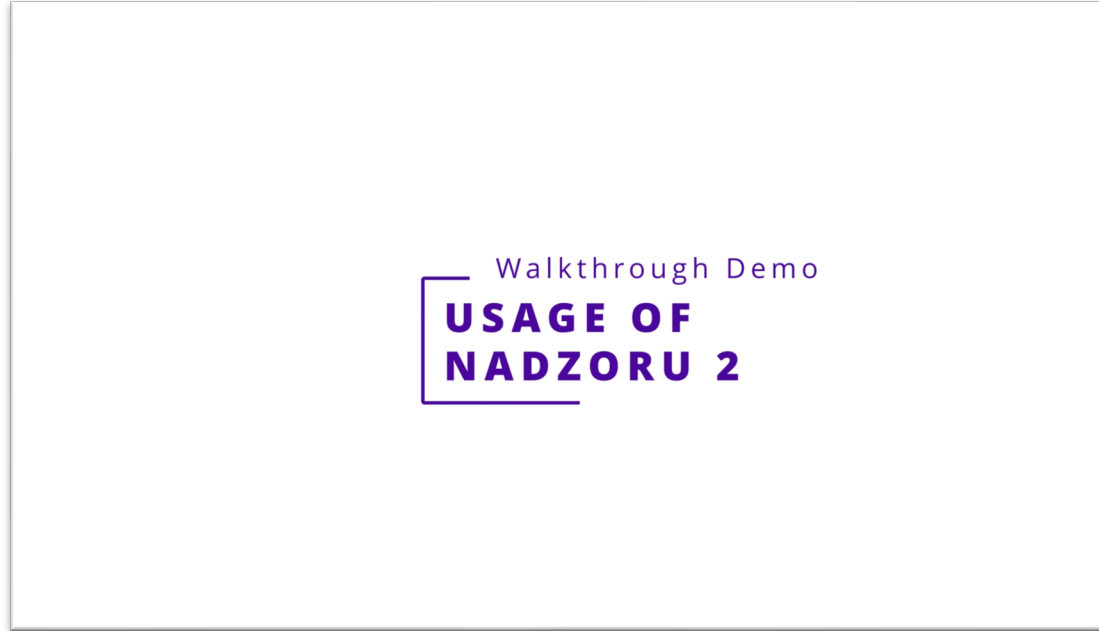
Using Nadzoru 2 to design the controller

Supervisory Control Theory (SCT)

- Free behaviours (G)
 - G1: Robot can (1) **move** to work, (2) **work**, (3) **move** to charge, and (4) **charge**
 - G2: Robot can **detect** whether it is inside the work or charge region
- Control specifications (E)
 - E1: The robot must move to work if it is not at the work region
 - E2: The robot must work when it is inside the work region
 - E3: The robot must move to charge if it is not at the charge region
 - E4: The robot must charge when it is inside the charge region



Nadzoru 2: walkthrough demo video



Video link: <https://www.youtube.com/watch?v=siRzF2sgUG4>

Using the generated code

The generated code consists of (1) the robot's control logic and (2) a virtual machine

Steps to integrate the code in C

1. Initialize the virtual machine

- `SCT_init()`

2. Implement callback functions for each event

- `check_atWork()`
 - Return a *boolean* indicating whether the uncontrollable event has triggered or not
- `callback_moveToWork()`
 - Execute the code associated with the action

3. Add the implemented callback functions to the virtual machine

- `SCT_add_callback()`

4. Call the virtual machine in the main loop

- `SCT_run_step()`

Using the generated code

1. Initialize the virtual machine

```
#include generic_mic.h

/* Initialize virtual machine */
SCT_init();
```

2. Implement callback functions for each event

```
/* Callback functions */
unsigned char check_atWork(void *data) {
    bool atWork = (x_coord > x_work) ? true : false;
    if (atWork) {
        printf("EVENT: atWork was triggered");
    }
    return atWork;
}

void callback_moveToWork(void *data) {
    printf("ACTION: moveToWork was triggered");
    move = true; direction = "work";
}
...
```


Using the generated code

3. Add the implemented callback functions to the virtual machine

```
/* Add callback functions associated with each event */  
SCT_add_callback(EV_atWork, NULL, check_atWork, NULL);  
SCT_add_callback(EV_moveToWork, callback_moveToWork, NULL, NULL);  
...
```

4. Call the virtual machine in the main loop

```
/* In mainloop, call the virtual machine */  
SCT_run_step();
```

Useful links

- Kilobot
 - [Wiki](#)
 - [Source code](#) for the example shown in this presentation
 - Nadzoru 2
 - [GitHub repository](#)
- ❑ Example scenario was adapted from G. Miyauchi, Mohamed S. Talamali, R. Groß, “[A comparative study of energy replenishment strategies for robot swarms](#)”, 14th International Conference on Swarm Intelligence (ANTS), pp. 3-15, LNCS vol 14987, 2024