Shape formation using Kilobots

A finite state machine approach

Sudhakar[1] Kishan[1] Eswara Srisai[1] Anurag[1]

[1]M.Tech scholar
Systems and Control Engineering

Indian Institute of Technology, Bombay
April 24, 2019

Overview

- Introduction
- Summary of first half of lab work
- Star planet orbiting
  - Orbiting
  - Escaping the close region
- Shape formation

Overview

- Introduction
- Summary of first half of lab work
- Star planet orbiting
  - Orbiting
  - Escaping the close region
- Shape formation

Kilobots

Specifications

- ATmega 328p processor
- Li-Ion 3.7V battery
- One IR transmitter-receiver pair
- One light sensor
- Two vibration motors (1 cm/sec, 45 degrees/sec)

Figure 1: Kilobot

About Kilobots [1]

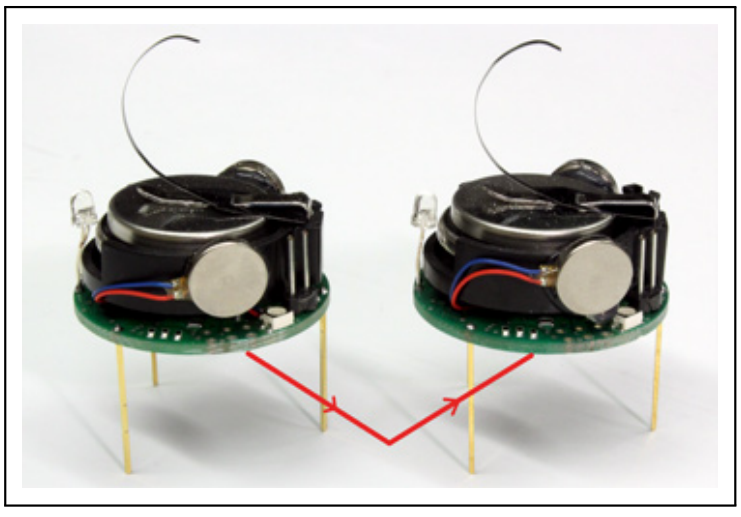


Figure 2: Communication between two Kilobots

- Reflecting IR light
- Communication up to 7 cm (32kb/s) away
- Using over-head controller

Wireless communication in Kilobots

- Sharing of same wireless channel by all robots
- CSMA-CA (Carrier Sense Multiple Access with Collision Avoidance) method [2].
- Reduction of channel bandwidth

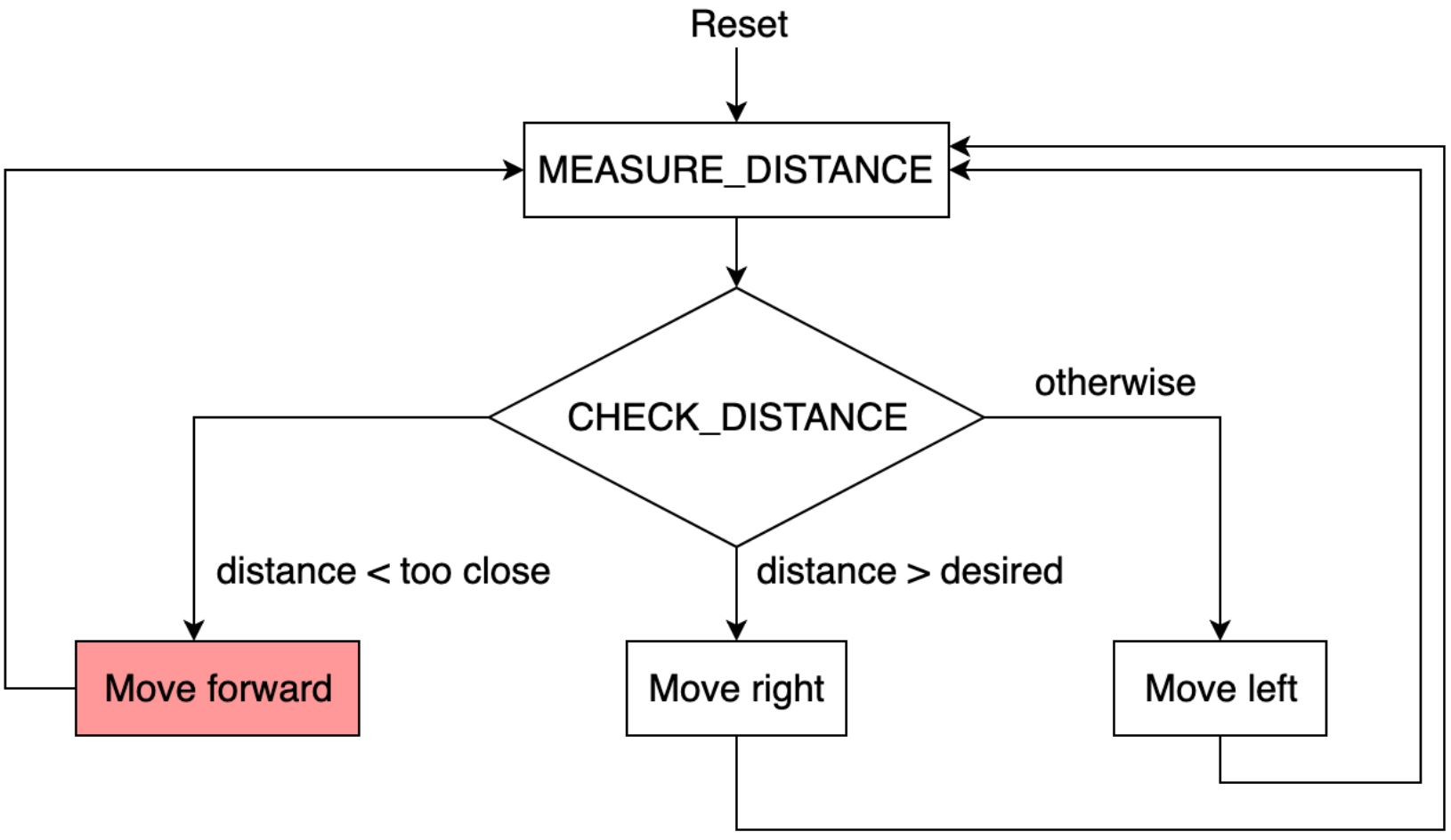
Overview

- Introduction
- Summary of first half of lab work
- Star planet orbiting
  - Orbiting
  - Escaping the close region
- Shape formation

Summary of first half of lab work

- Familiarization with Kilobots
- Establishing communication between two Kilobots (speaker and listener)
- Implementing naive algorithm for orbiting of Kilobots (star and planet)
- Moving towards the direction of light source
- Synchronizing phase of blinking LEDs

Flowchart


Reset
MEASURE_DISTANCE
CHECK_DISTANCE
otherwise
distance < too close
distance > desired
Move forward
Move right
Move left

Naive orbiting algorithm

Demonstration

Figure 3: Naive orbiting algorithm

Overview

- Introduction
- Summary of first half of lab work
- Star planet orbiting
  - Orbiting
  - Escaping the close region
- Shape formation

Efficient orbiting using a Finite State Machine (FSM)

Objective: Algorithm to allow a planet to orbit n stars from any initial condition.

- **Stars:** Stationary bots around which planet rotates (Black)

Figure 4: Single star system

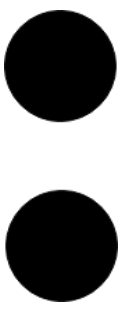


Figure 5: Linear star system

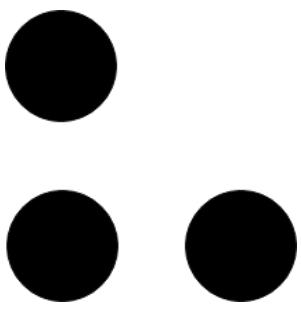


Figure 6: Triangular star system

Efficient orbiting using a Finite State Machine (FSM)

Objective: Algorithm to allow a planet to orbit n stars from any initial condition.

- **Stars:** Stationary bots around which planet rotates (Black)
- **Planet:** Dynamic bots rotating around stars (Gray)

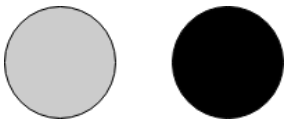


Figure 4: Single star system

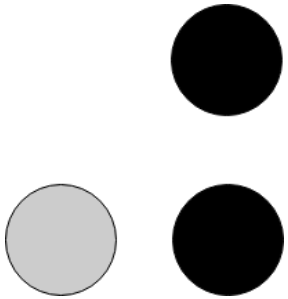


Figure 5: Linear star system

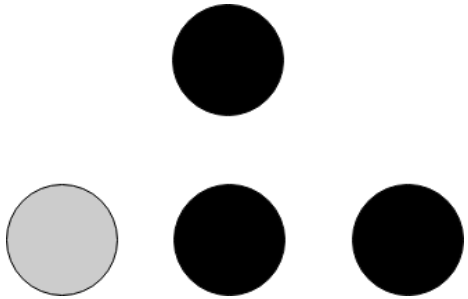
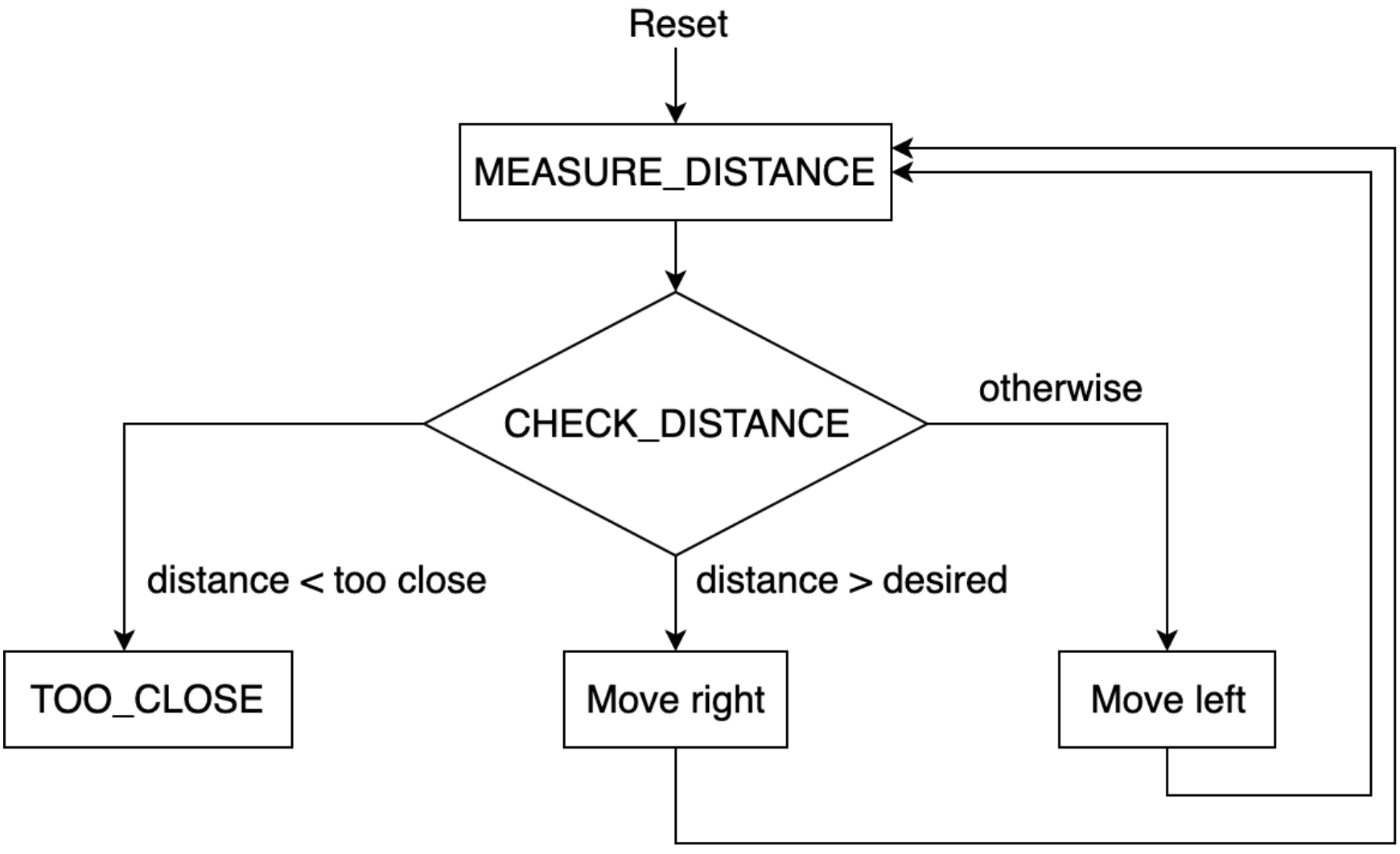


Figure 6: Triangular star system

Flowchart


Reset
MEASURE_DISTANCE
CHECK_DISTANCE
otherwise
distance < too close
distance > desired
TOO_CLOSE
Move right
Move left

Efficient orbiting using FSM

Demonstration

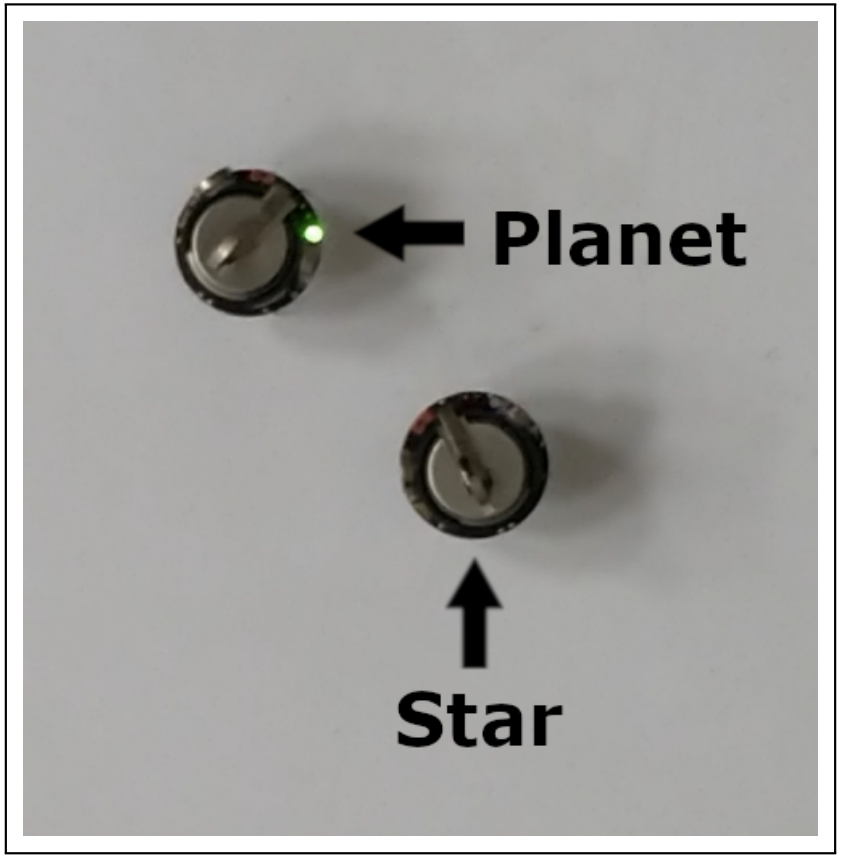


Figure 7: Efficient star-planet orbiting using single communication

Escaping the close region

Objective: Designing a robust algorithm to reach the desired orbit distance without hitting the star.

Escaping the close region

Objective: Designing a robust algorithm to reach the desired orbit distance without hitting the star.

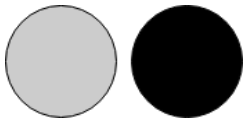


Figure 8: Planet too close to star

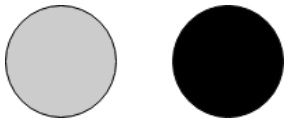


Figure 9: Desired distance of orbit

Snapshots

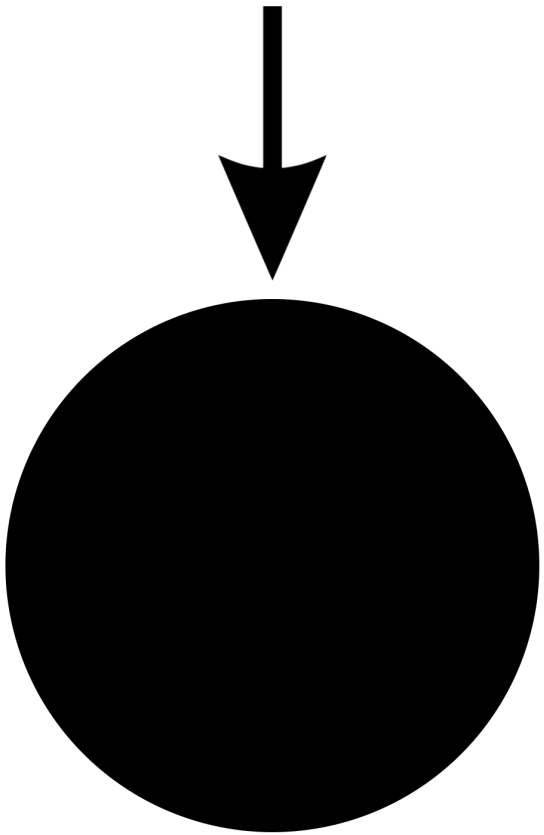


Figure 10: Snapshots of working algorithm

Snapshots

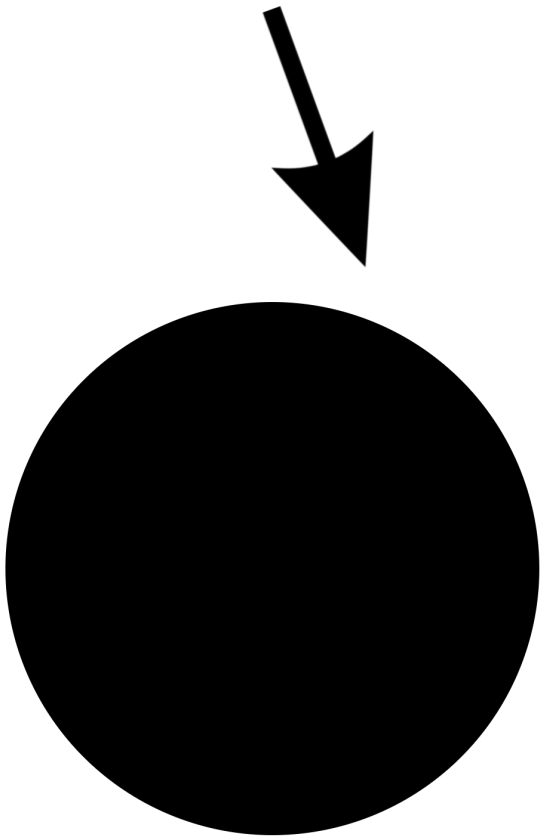


Figure 10: Snapshots of working algorithm

Snapshots

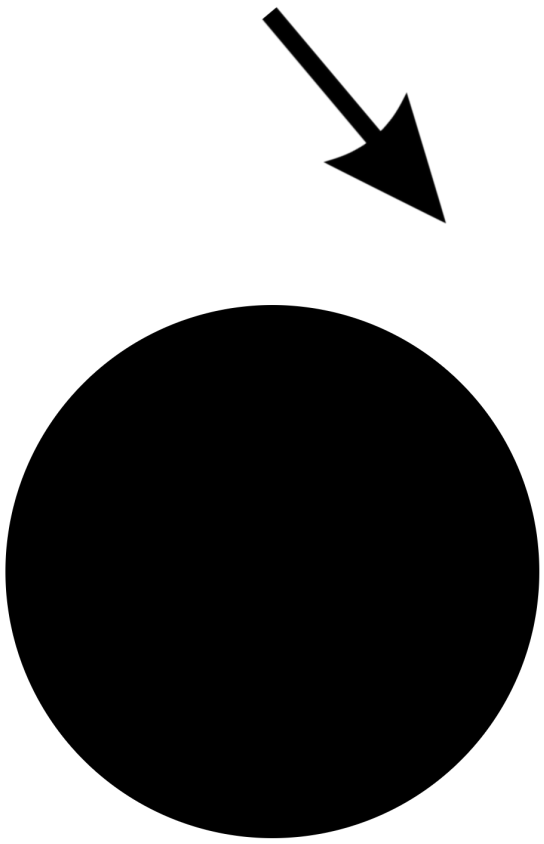


Figure 10: Snapshots of working algorithm

Snapshots

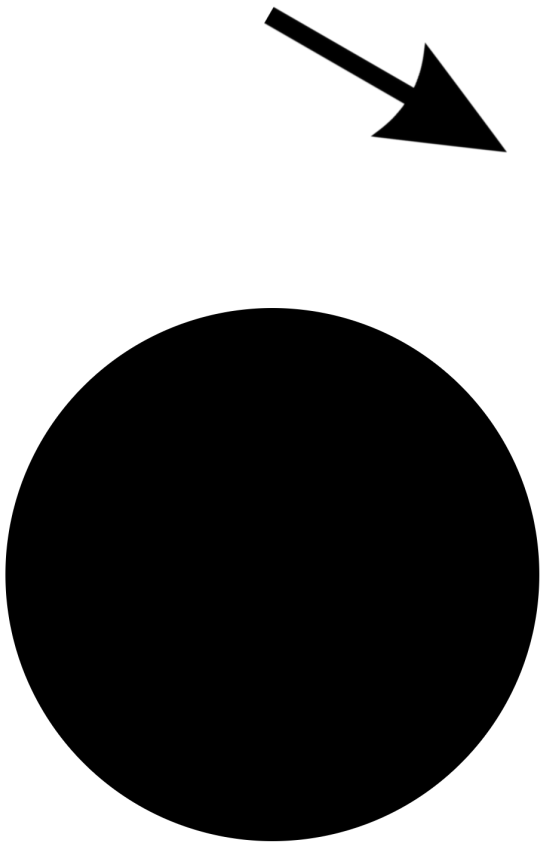


Figure 10: Snapshots of working algorithm

Snapshots

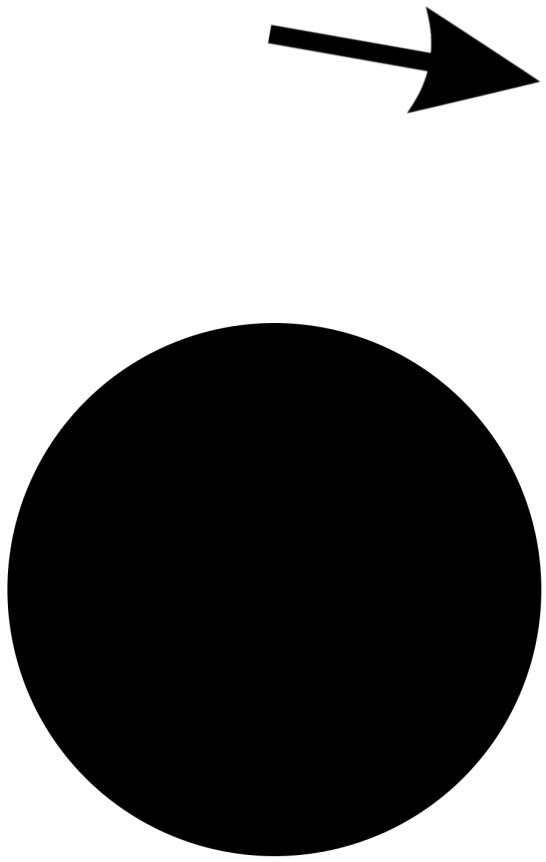


Figure 10: Snapshots of working algorithm

Snapshots

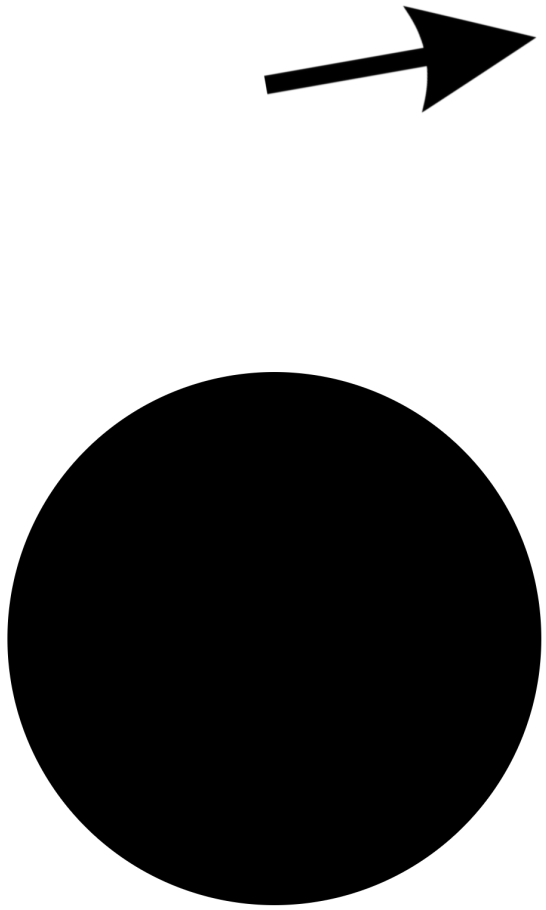


Figure 10: Snapshots of working algorithm

Snapshots

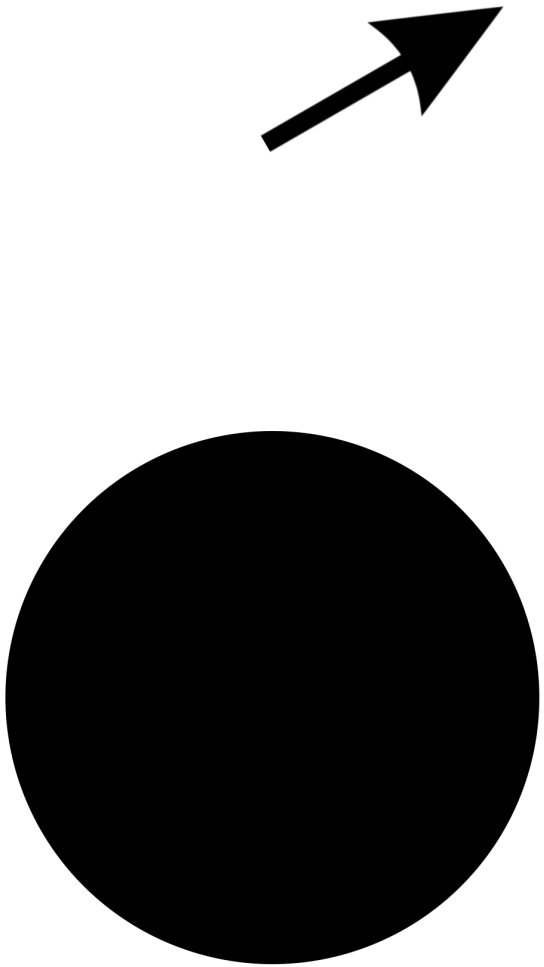


Figure 10: Snapshots of working algorithm

Snapshots

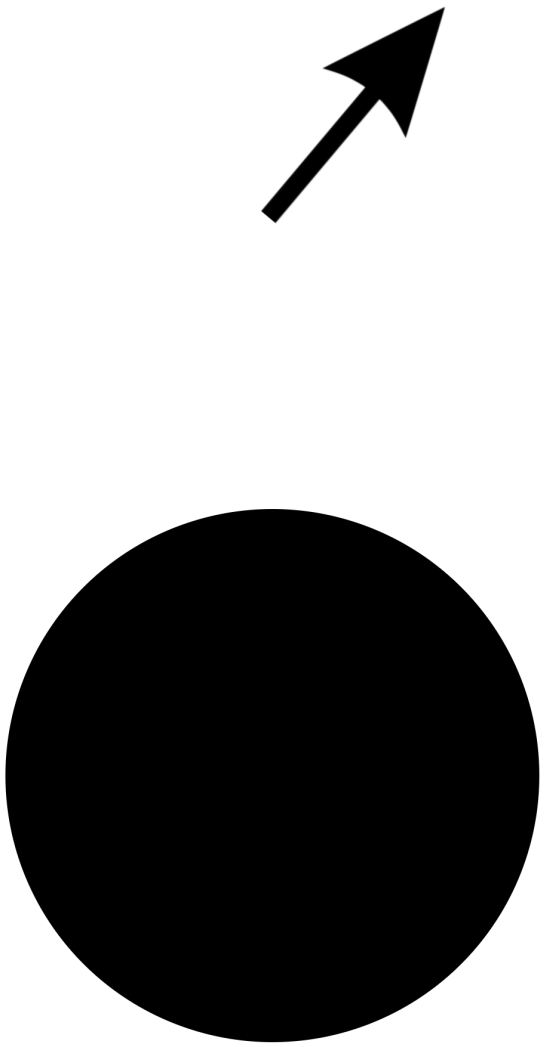


Figure 10: Snapshots of working algorithm

Snapshots

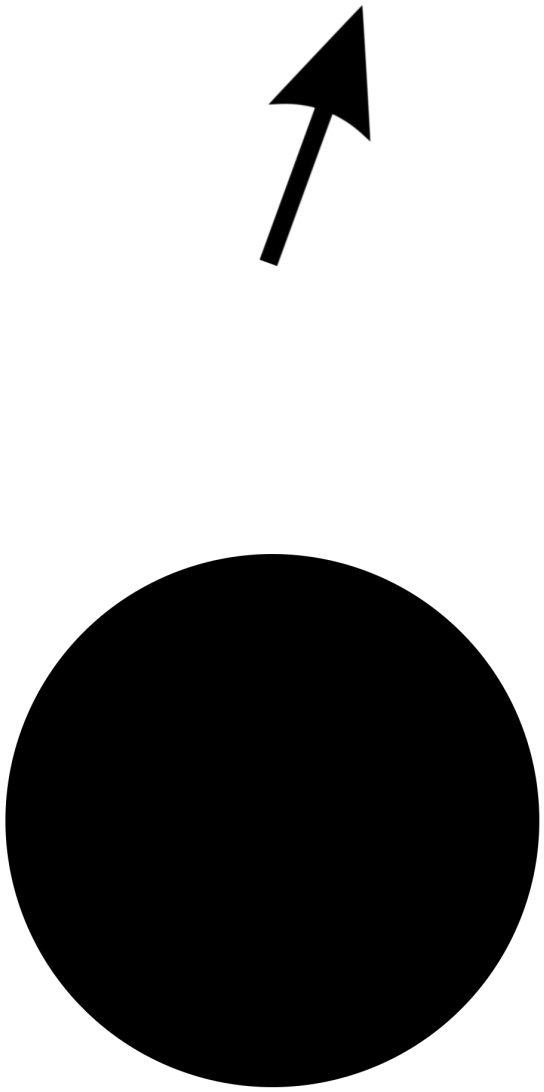
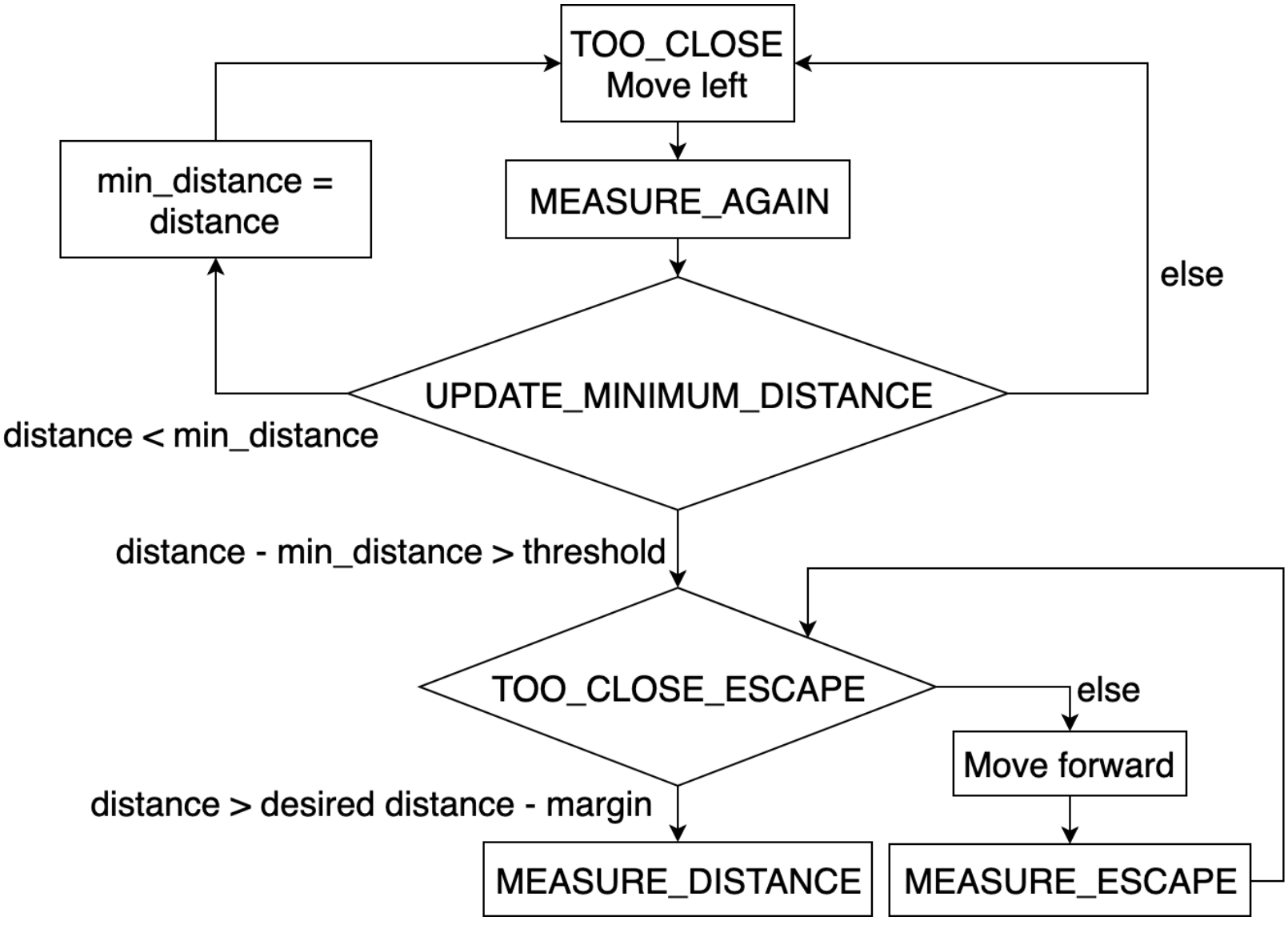


Figure 10: Snapshots of working algorithm

Flowchart


TOO_CLOSE
Move left
MEASURE_AGAIN
min_distance =
distance
else
UPDATE_MINIMUM_DISTANCE
distance < min_distance
distance - min_distance > threshold
TOO_CLOSE_ESCAPE
else
Move forward
distance > desired distance - margin
MEASURE_DISTANCE
MEASURE_ESCAPE

Efficient orbiting using FSM

Demonstration

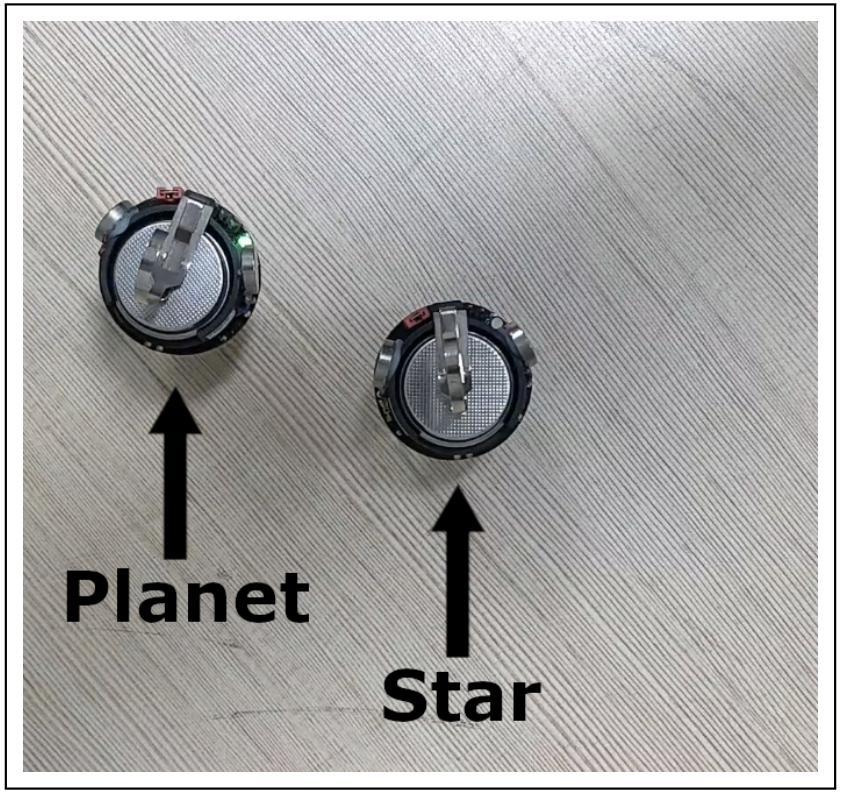


Figure 11: Escaping too close region of star by planet followed by orbiting

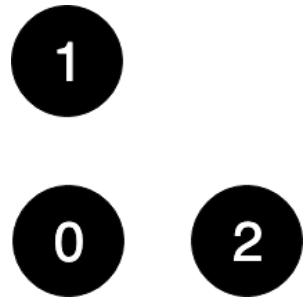
Overview

- Introduction
- Summary of first half of lab work
- Star planet orbiting
  - Orbiting
  - Escaping the close region
- Shape formation

Shape formation

Objective: Distributed algorithm to generate a desired shape [3].

- **Guides:** Index $0, 1, 2$ acts as reference for coordinate axis by continuously transmitting their index.

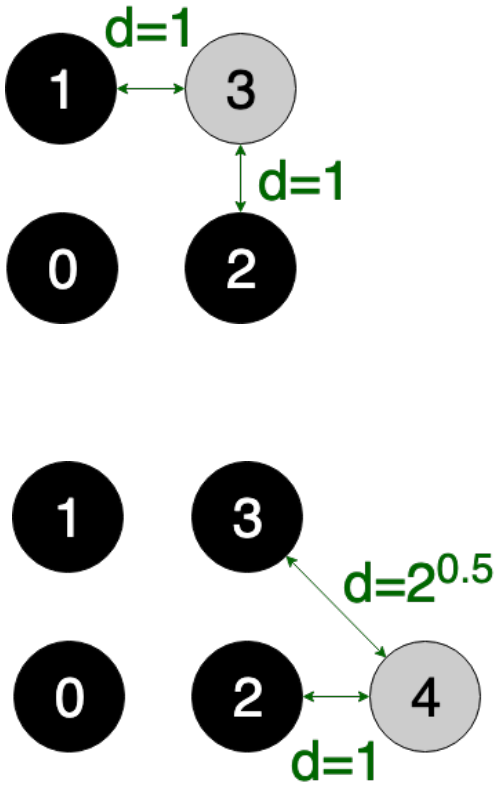


Shape formation

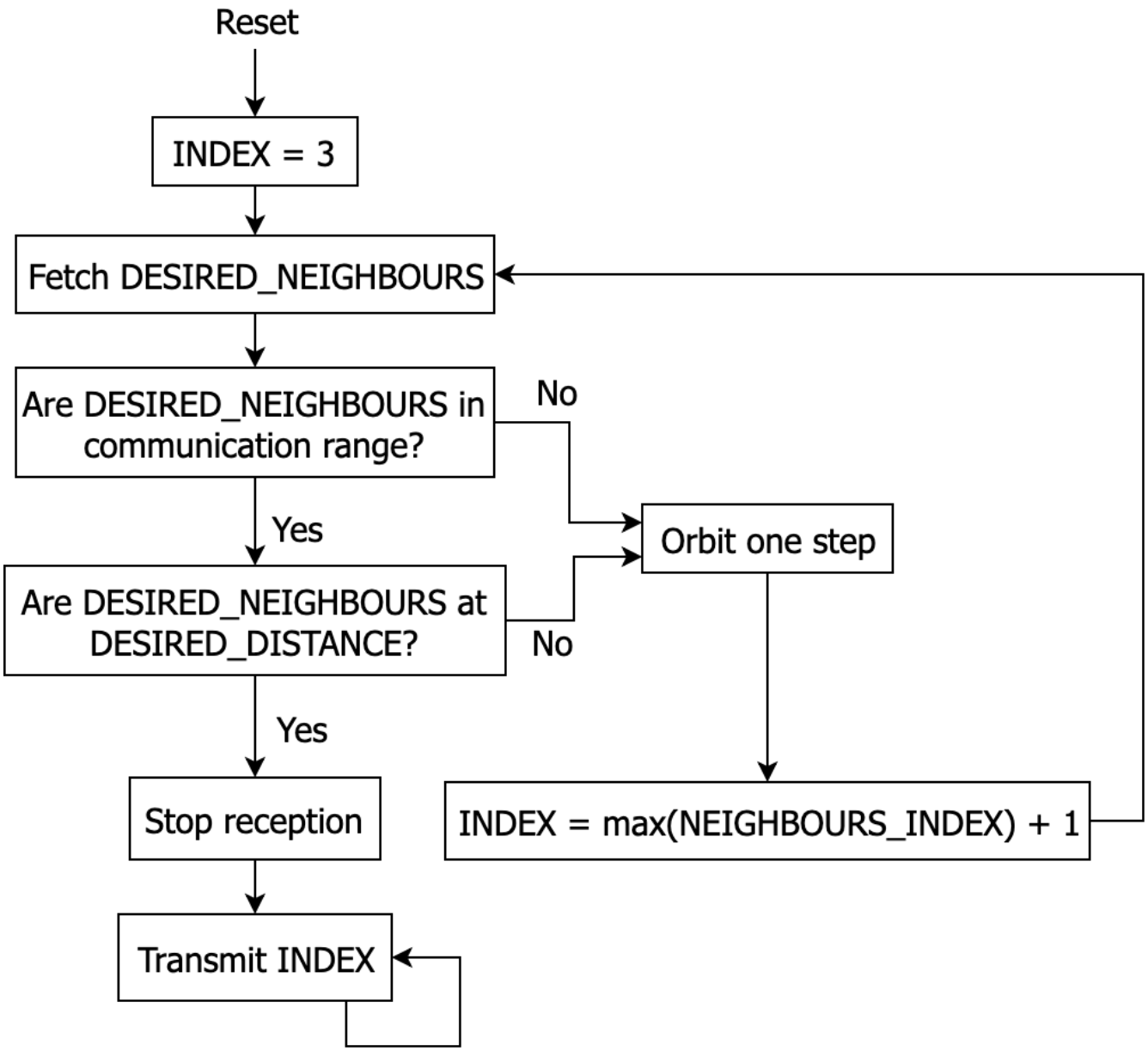
- **Builders:** Index 3 onwards for shape formation
- For forming a linear shape of width 2, the **shape matrix** would look like

$$\begin{bmatrix} \textit{Index} & N_1 & DD_1 & N_2 & DD_2 \\ \hline 3 & 1 & 1 & 2 & 1 \\ 4 & 2 & 1 & 3 & \sqrt{2} \\ 5 & 3 & 1 & 4 & 1 \\ \dots & \dots & \dots & \dots & \dots \end{bmatrix}$$

where,
N_i : Desired neigbour i
DD_i: Desired distance from neighbour i.



Flowchart


Reset
INDEX = 3
Fetch DESIRED_NEIGHBOURS
Are DESIRED_NEIGHBOURS in communication range?
No
Yes
Orbit one step
Are DESIRED_NEIGHBOURS at DESIRED_DISTANCE?
No
Yes
Stop reception
INDEX = max(NEIGHBOURS_INDEX) + 1
Transmit INDEX

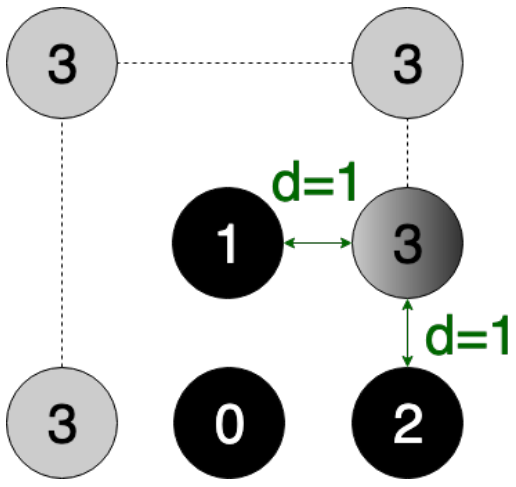


Figure 12: First builder

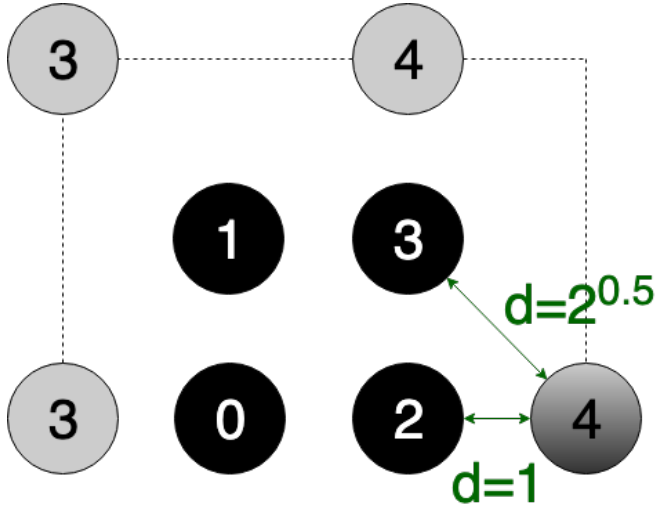


Figure 13: Second builder

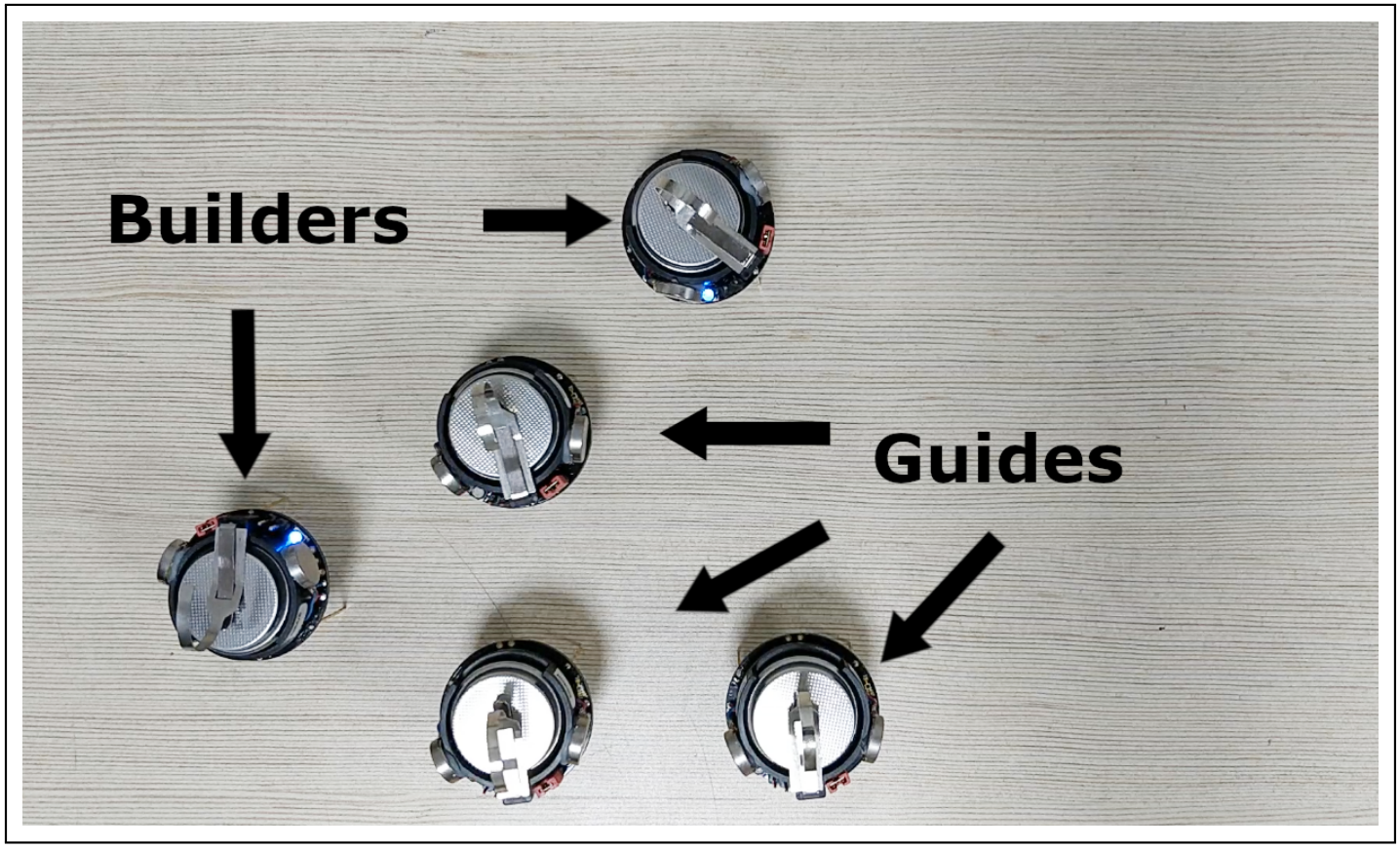


Figure 12: Rectangle shape formation by Kilobots (l=3, b=2)

Conclusion

Challenges

- Calibration of Kilobots
- Non-smooth surface

Scope

- Integration of individual building blocks
- Optimization based localization scheme [3]
- Macros for generating shape matrix.

Reference

K Team. *Kilobot user manual*. URL: https://ftp.k-team.com/kilobot/user_manual/Kilobot_UserManual.pdf.

Vangie Beal. *CSMA/CD - Carrier Sense Multiple Access / Collision Detection*. URL: https://www.webopedia.com/TERM/C/CSMA_CD.html.

Michael Rubenstein, Alejandro Cornejo, and Radhika Nagpal. "Programmable self-assembly in a thousand-robot swarm". In: *Science* 345.6198 (2014), pp. 795–799. ISSN: 0036-8075. DOI: 10.1126/science.1254295. eprint: https://science.sciencemag.org/content/345/6198/795.full.pdf. URL: https://science.sciencemag.org/content/345/6198/795.