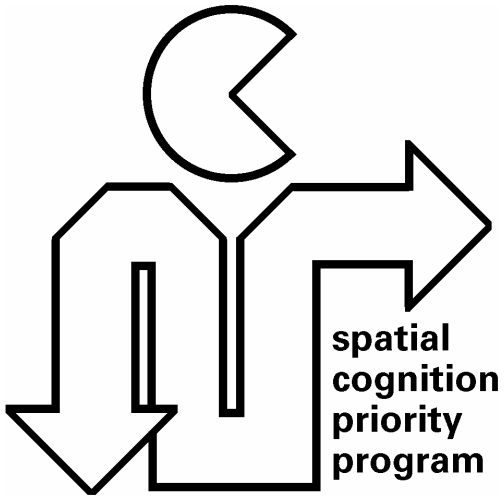
# Architecture and Applications of the Bremen Autonomous Wheelchair

**Thomas Röfer and Axel Lankenau**




Bremen Institute of Safe Systems
Center of Computing Technology

University of Bremen

# Outline

## The Bremen Autonomous Wheelchairs

## Safety in Robotics

- Formal Design Approach
- Fault Tree Based Hazard Analysis
- Specification of the Environment
- Architecture
- SAM

## Applications

- Adaptive Speed Control
- Obstacle Avoidance
- Local Metrical Navigation
- Basic Behaviors
- Route Navigation

## Conclusion

# The Bremen Autonomous Wheelchairs

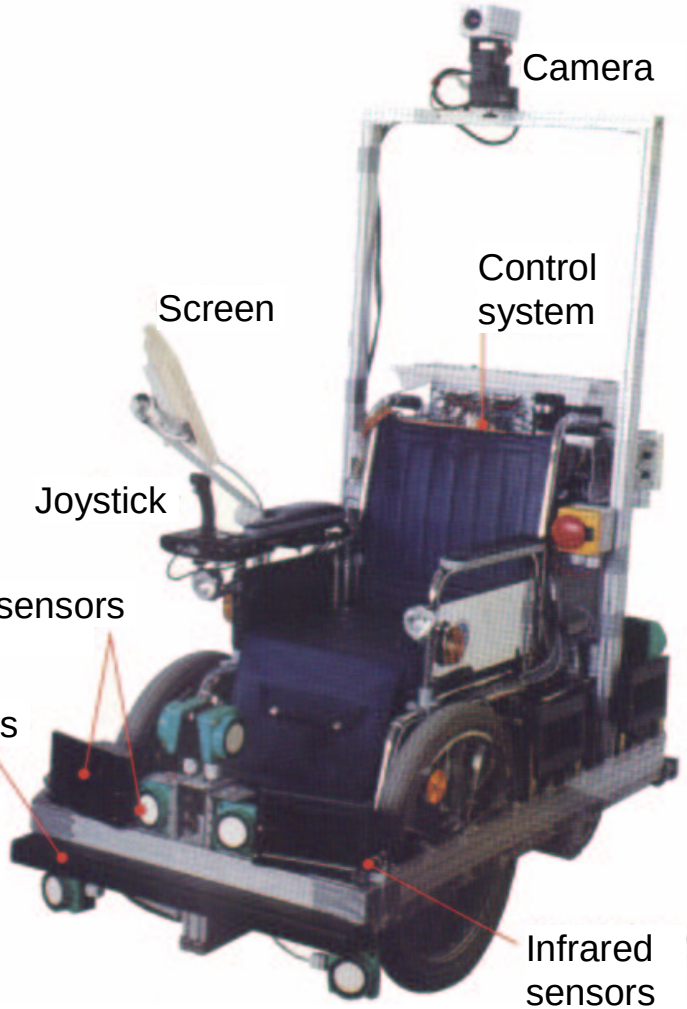
Laser scanner

Camera

Speech input

Ultrasonic sensors

Infrared sensors



Camera

Screen

Control system

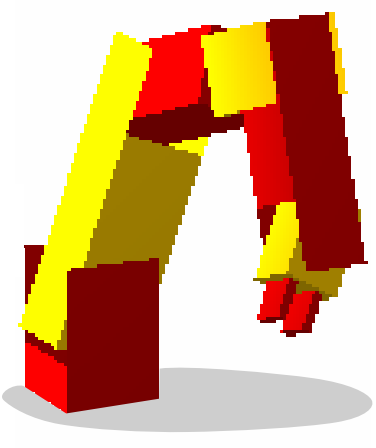
Joystick

Sonar sensors

Bumpers

Infrared sensors

# Safety in Robotics



Industrial robotics

Service robotics

Rehabilitation robotics

**Increasing safety demands**

# Formal Design Approach

## Hazard Analysis

- Fault tree analysis
- Specification of undesired system behavior

## Derivation of Safety Requirements

- Specification of the environment
- Specification of safety properties

## Definition of Safety Mechanisms

- Controller ensuring system safety
- Potential introduction of new hazards caused by the controller

## Verification of Safety Properties

# Fault Tree Based Hazard Analysis

## Fault Tree Segment (Problems of External Sensors)

```
X Failure of external sensors
  | X.1 Measuring error that may cause a collision
       & X.1.1 Too large values measured by sensors
            | X.1.1.1 Too large values measured up
                      to n consecutive times.
            | X.1.1.2 Too large values measured more
                      than n consecutive times #
       & X.1.2 Obstacle distances overestimated.
  | X.2 Disastrous breakdown of external sensors
       & X.2.1 No distances measured #
       & X.2.2 Breakdown not detected.
  | X.3 Obstacle not detectable by external sensors #
```

# Specification of the Environment

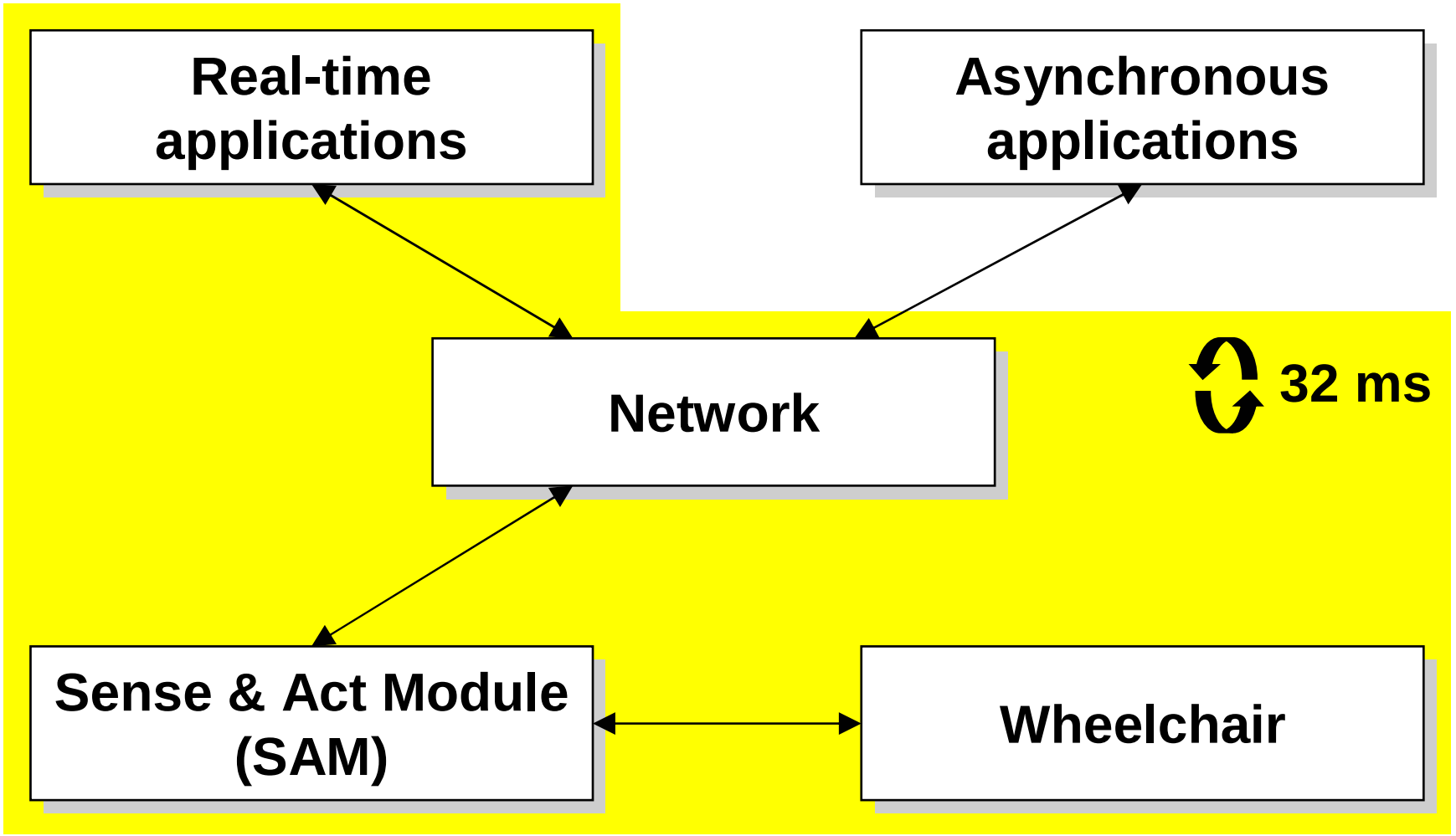
## Fault Tree Leaves:

```
Safety Requirements of the system not satisfied
| 1 Collision (at sensor level)
    | 1.1 Passive collision #
    | 1.2 Active collision
    ...
| 2 Crash (not at sensor level) #
| 3 Downfall #
...
```

## Requirements Imposed on the Environment:

- No “active” obstacles
- Maximum horizontal extent of every obstacle at sensor level
- No stairs etc.

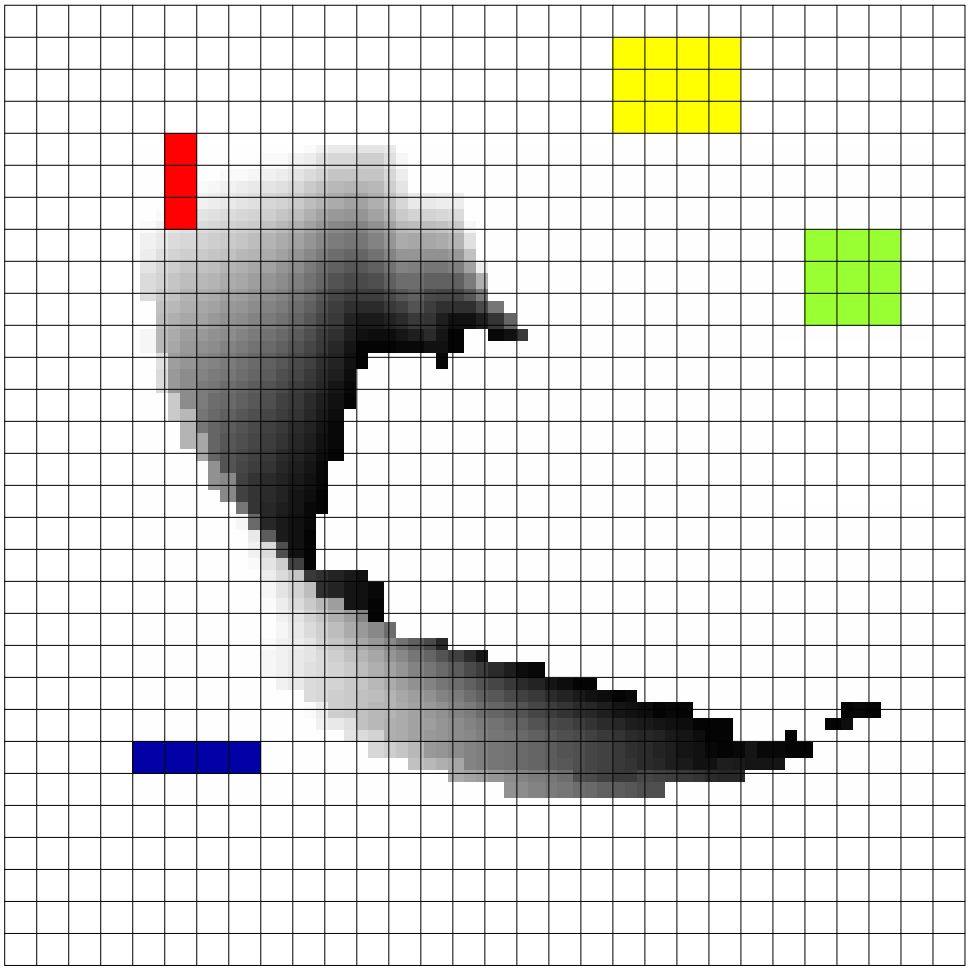
# Architecture



# SAM

## Local Obstacle Map

- Stores the local surroundings of the wheelchair
- Is shifted analogously to the movement of the system
- “Aging” of measurements
- Cell size 3 x 3 $cm^2$
- Resolution $120^2$ cells
- Update < 3 ms



## Virtual Sensors

- Depending on rotation, direction, and steering
- Anticipation of collisions
- Distance calculation < 75 µs

# Applications

**Route navigation**

**Basic behaviors**

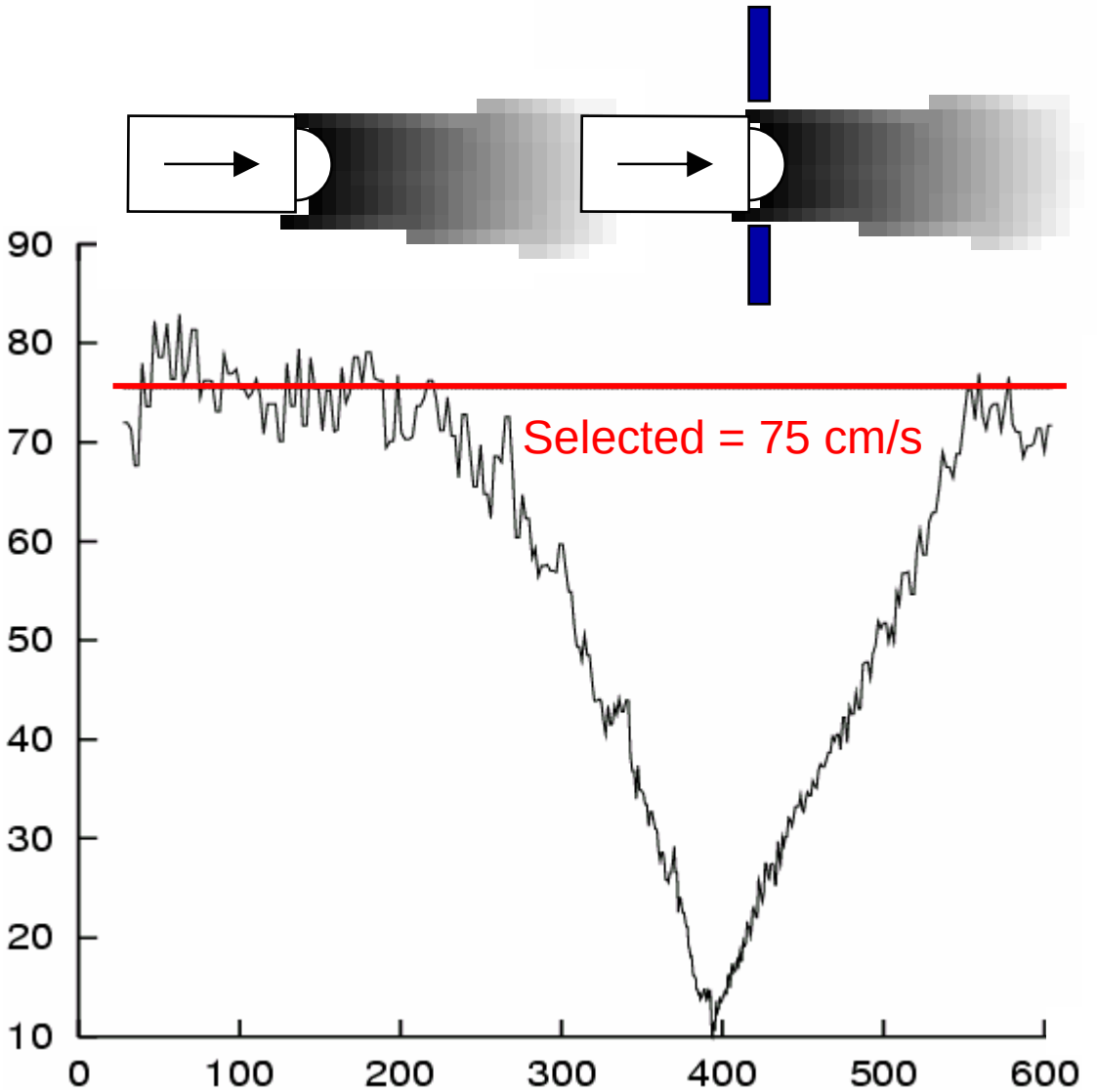
**Local metrical navigation**

**Obstacle avoidance**

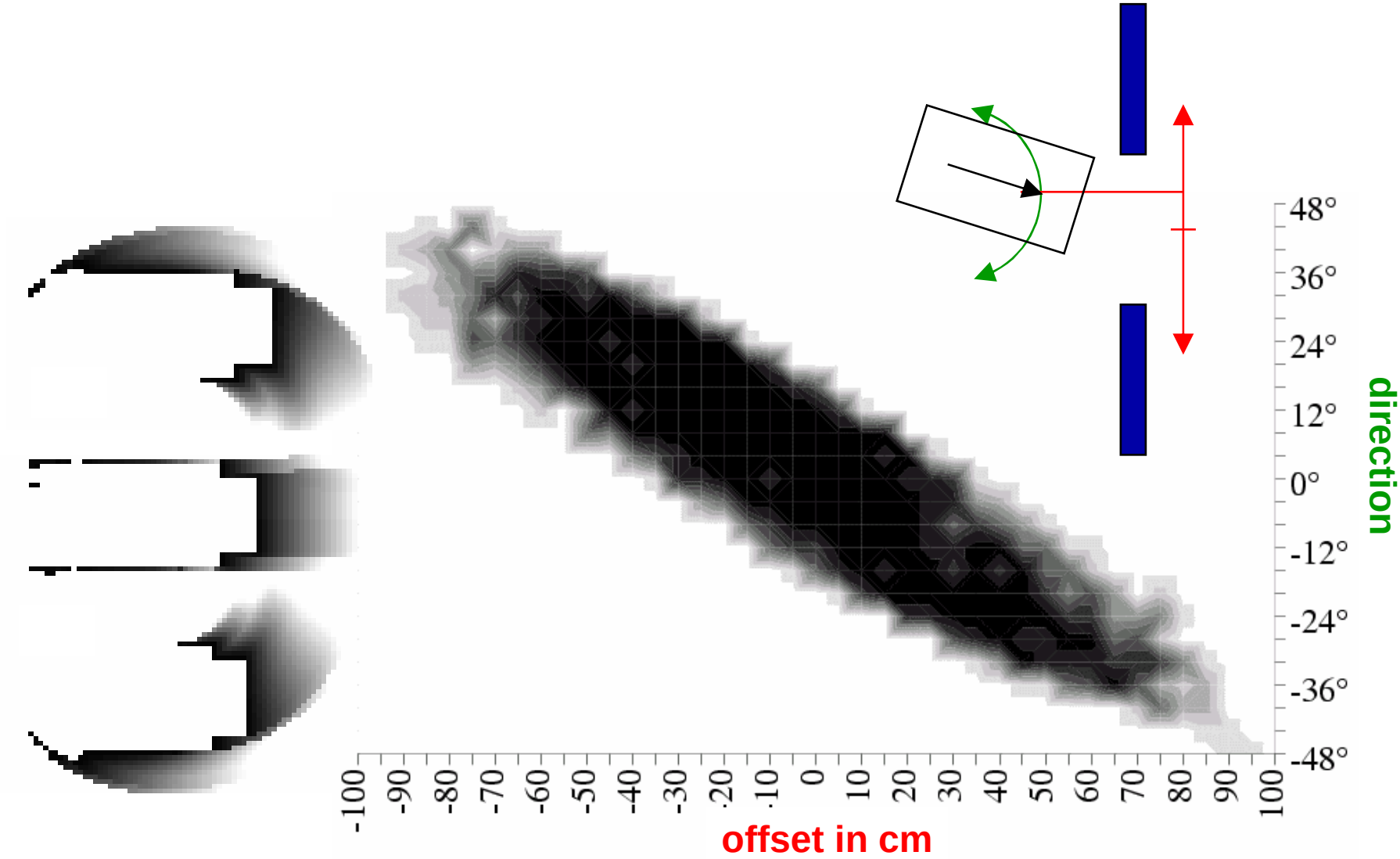
**Adaptive speed control**

**Sense & Act Module (SAM)**

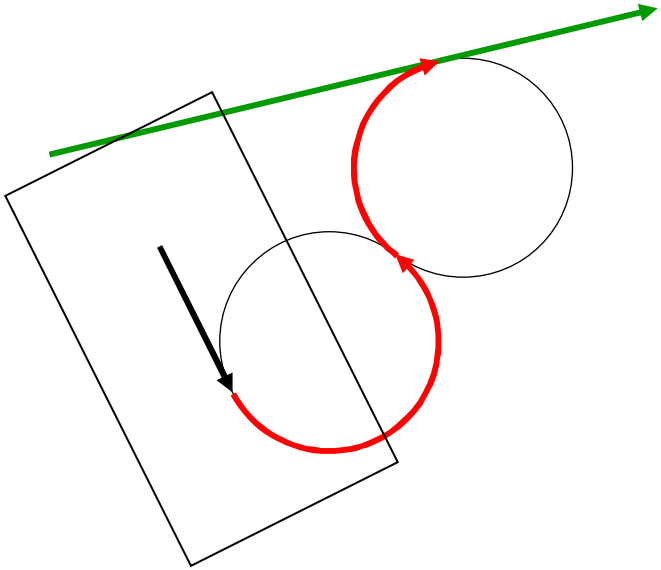
# Adaptive Speed Control

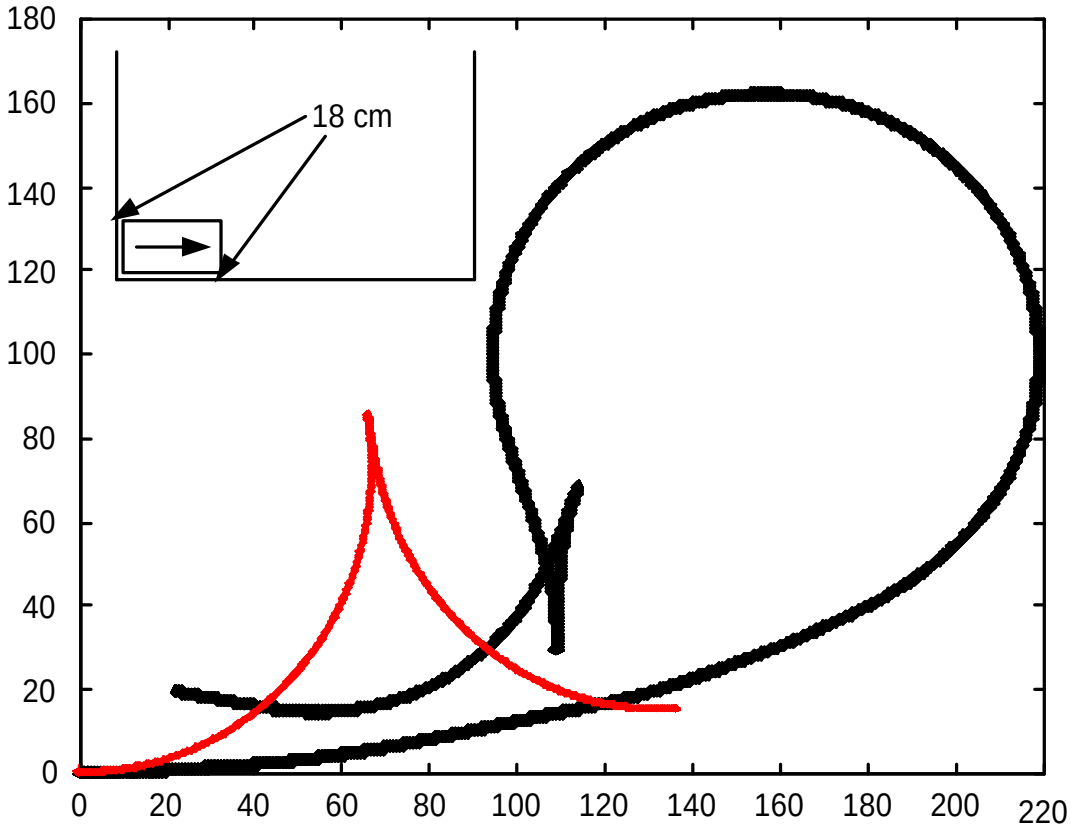


# Obstacle Avoidance



# Local Metrical Navigation





# Basic Behaviors

## Forwards and Backwards

- Corridor-following
- Wall-following left/right

## Only Forwards

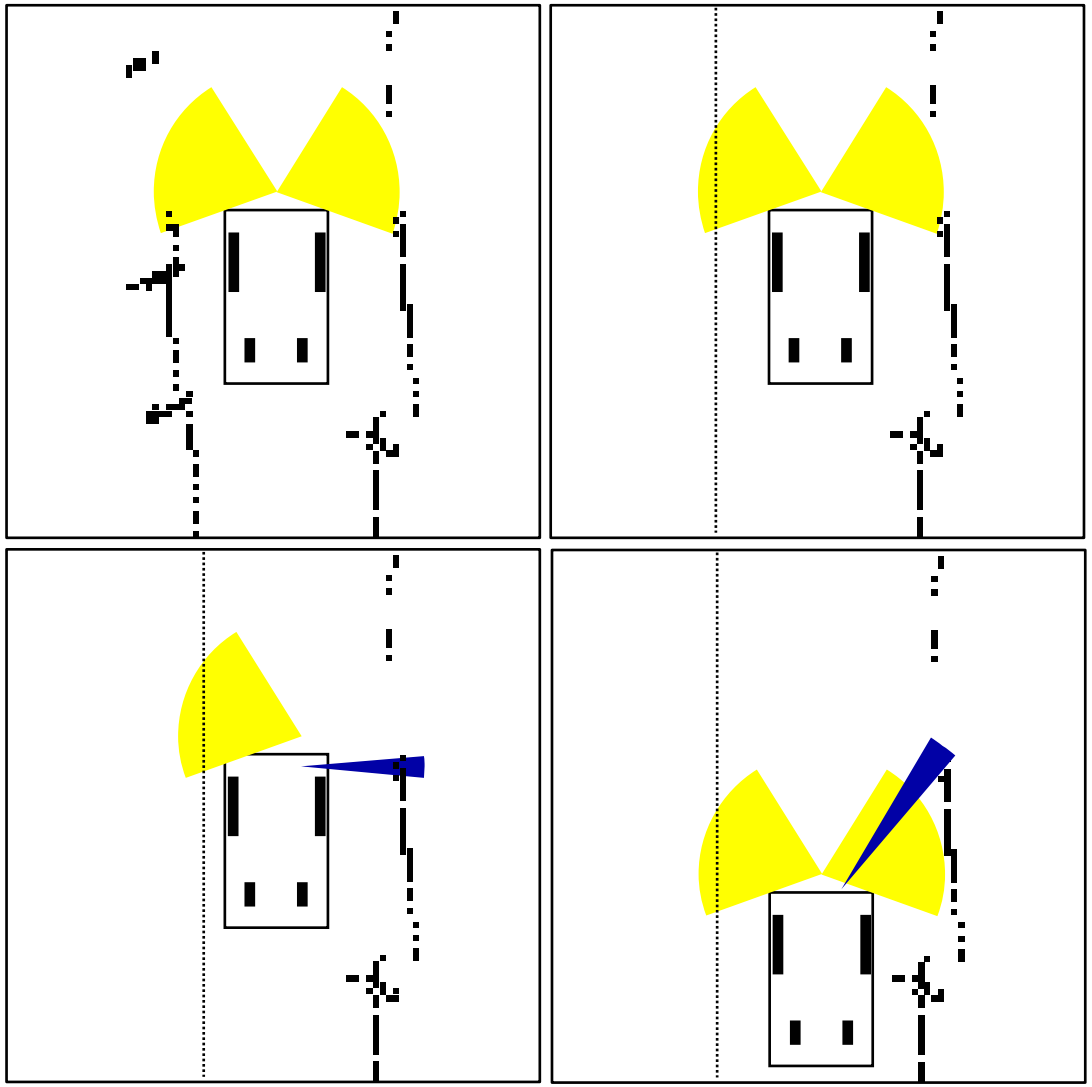
- Turning into the left/right door

## Automatically

- Turning round

## Miscellaneous

- Stop



# Route Navigation

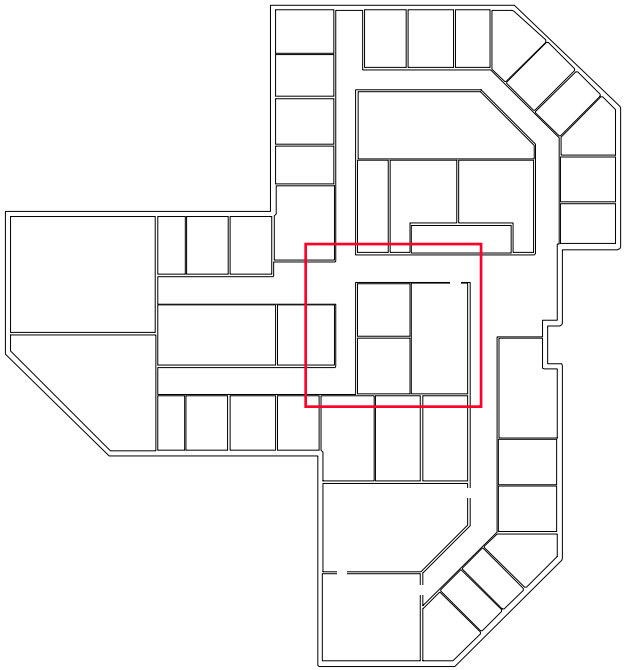
## Behaviors

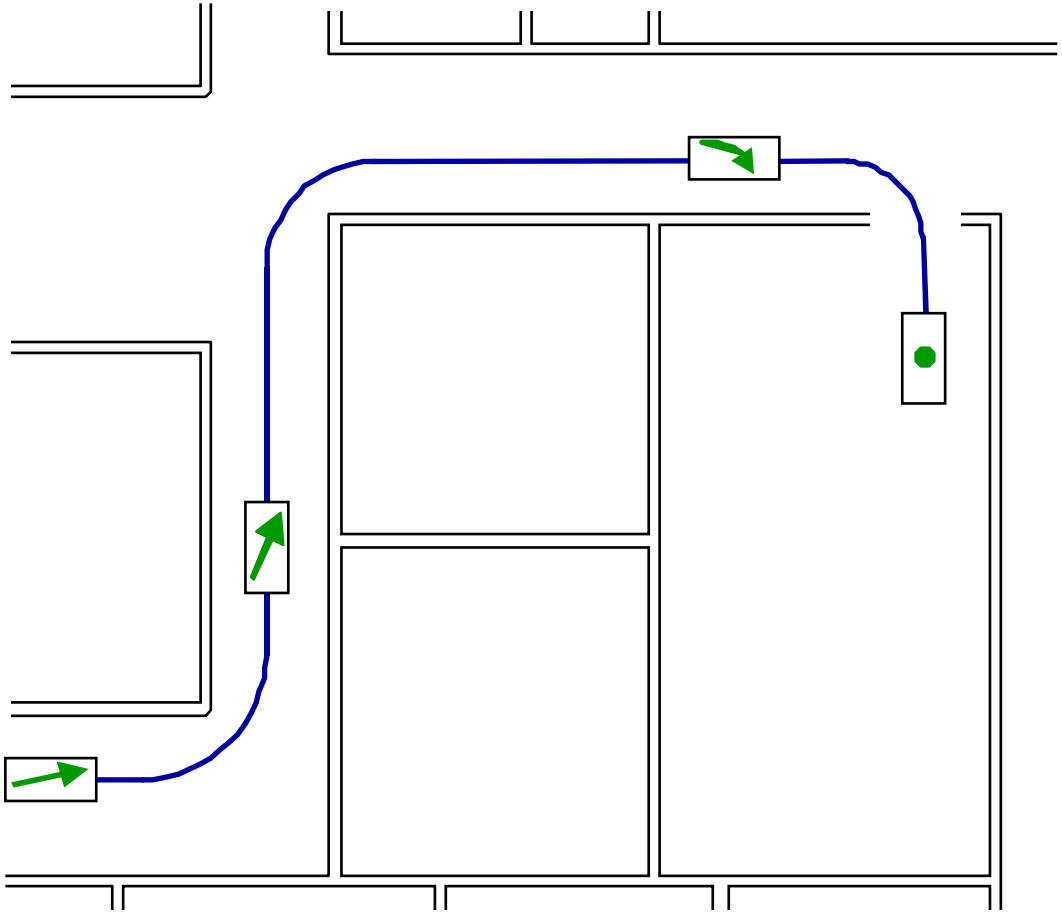

• Wall-following left


• Wall-following right

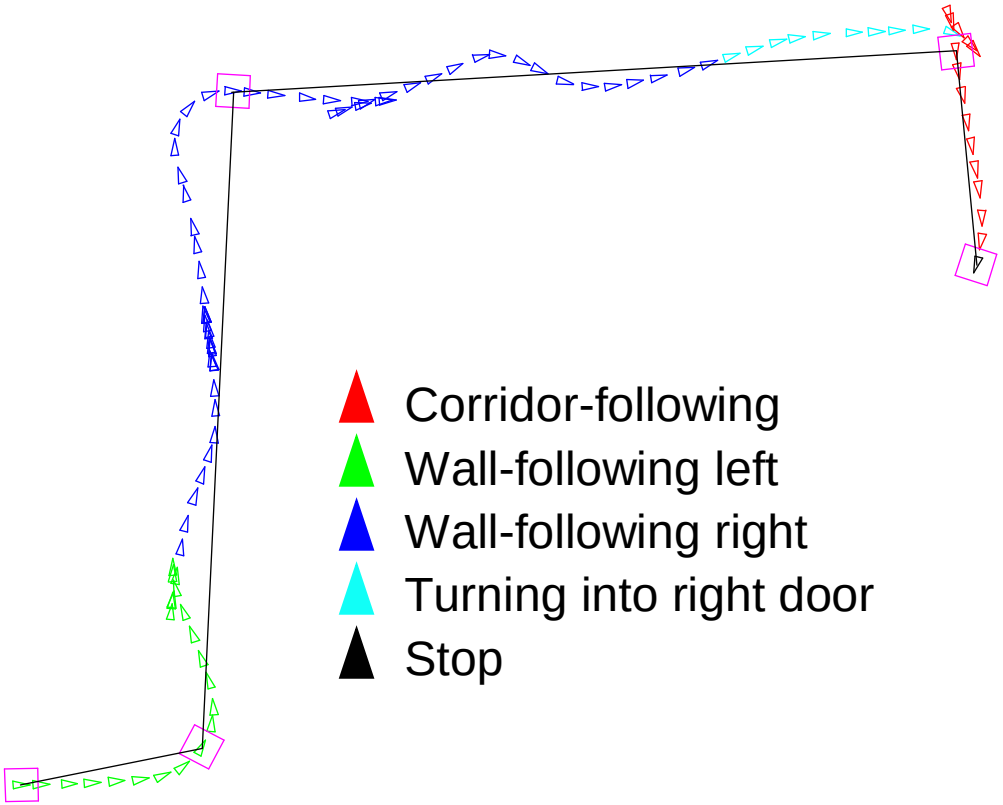

• Turning into right door


• Stop

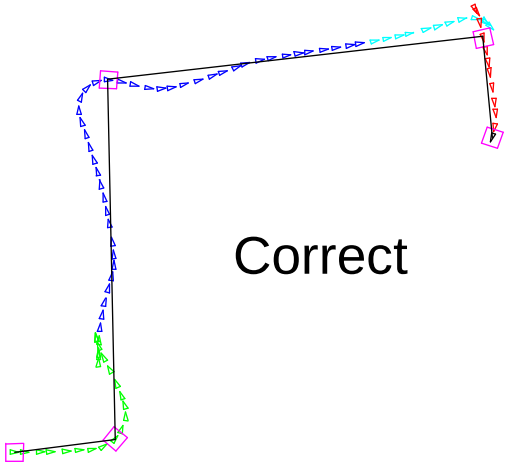




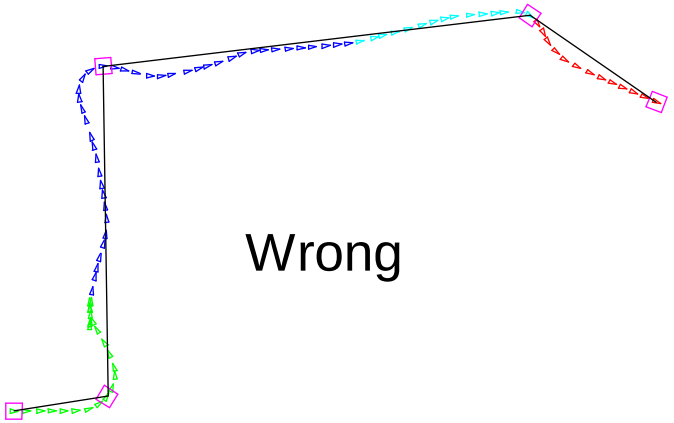
# Route Descriptions



Corridor-following
Wall-following left
Wall-following right
Turning into right door
Stop



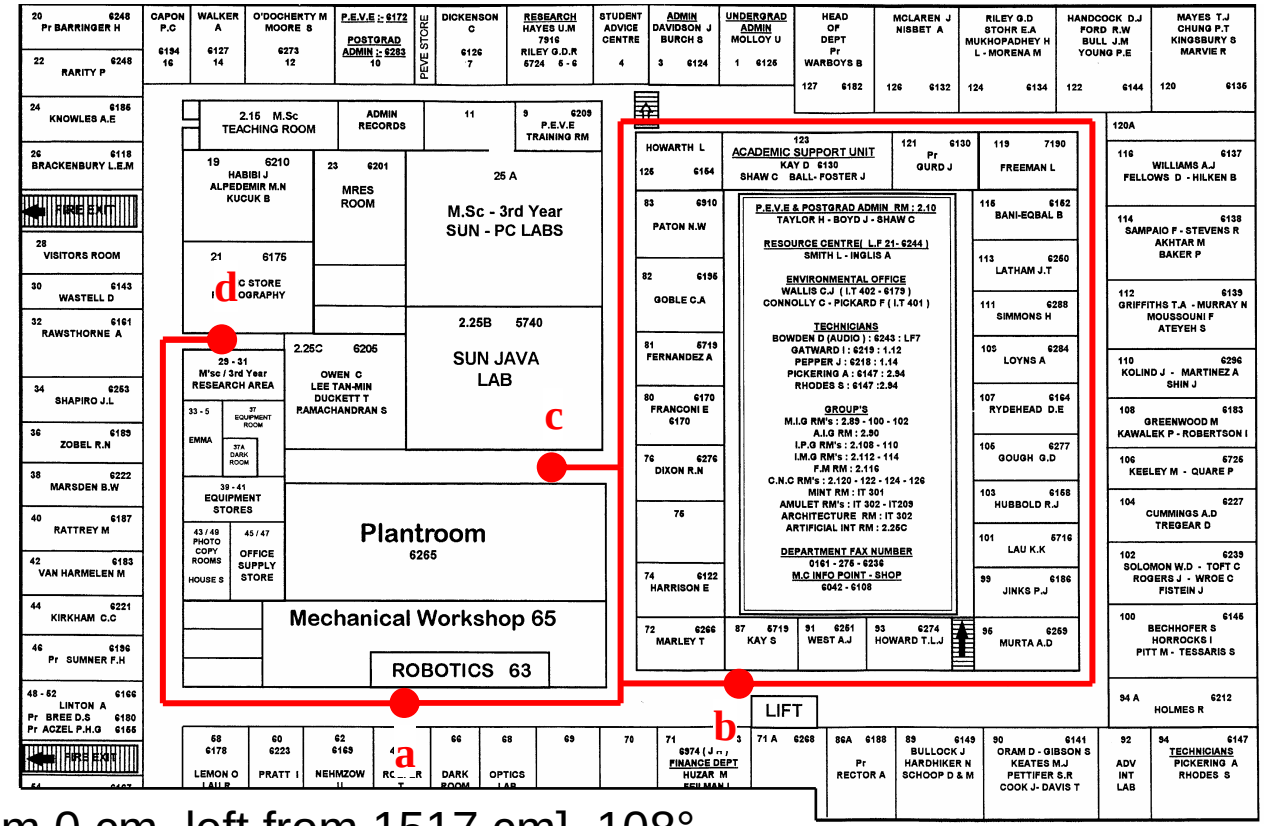
Correct



Wrong

224cm, 75°, 799cm, -83°, 880cm, -87°, 260cm

# Results




## Example b-c

- 1963 cm [right from 0 cm, left from 1517 cm], 108°,
- 5474 cm [right from 2453 cm, left from 4748 cm], 102°,
- 3215 cm [right from 647 cm, left from 2895 cm], 98°,
- 2983 cm [right from 539 cm], 83°,
- 516 cm [stop at 448 cm]

# Outlook

## Formal Verification of “Robotic Issues”

- Problem: Modeling of the environment is very complex
- Automated Testing

## Enhanced Support for Handicapped Users

- Commands by speech recognition
- Further basic behaviors, e.g. docking to a table

## Advanced Navigation

- Integration of several routes to “route maps”
- Outdoor navigation