TECHNOLOGICAL EDUCATIONAL INSTITUTE OF CENTRAL MACEDONIA
DEPARMENT OF INFORMATICS & COMMUNICATIONS

Master of Science in Communication & Information Systems

Localization and Obstacle Avoidance of an autonomous mobile robot based on dead reckoning and range data

Gkirinis Athanasios

June 2014 Serres

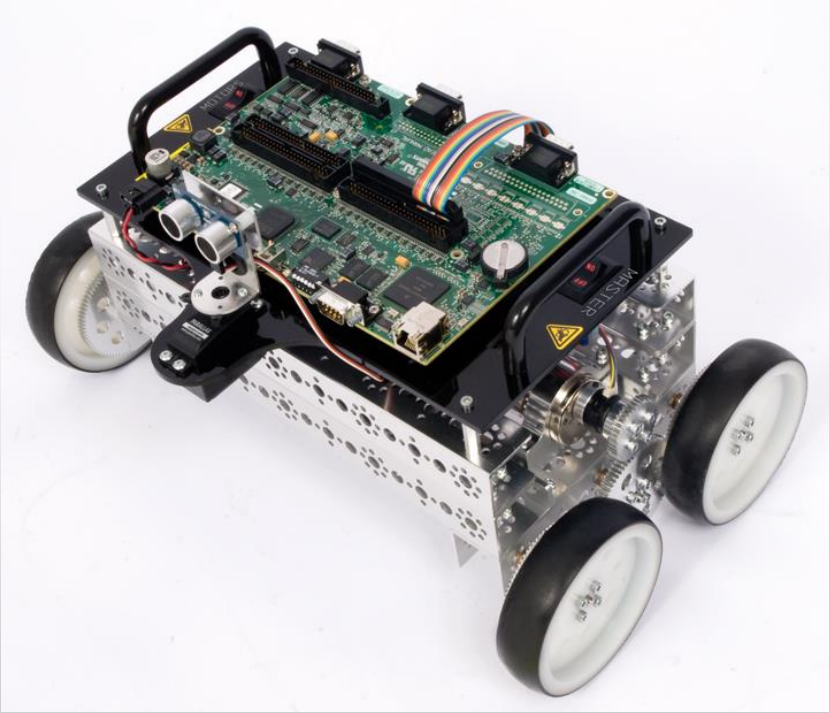
Problem: *Autonomous vehicles are the heart of next generation automotive technology and a lot of effort is dedicated to the development of appropriate sensors and algorithms for autonomous navigation, obstacle avoidance and environmental mapping*. The problem is to optimize obstacle avoidance in indoors environment and measure the robot pose as accurately as possible.

Objectives: The purpose of this thesis is to use appropriate navigation algorithms and program the robot's embedded controller for real-time localization and obstacle avoidance.

Methodology: We have used a robotic platform manufactured by National Instruments Corporation. The experimental robotic platform is equipped with a supersonic sensor for range measurements and with dead reckoning sensors, like optical encoders for odometry measurements. For obstacle avoidance we have used the Vector Field Histogram Algorithm and for the Localization of the robot the dead reckoning method.

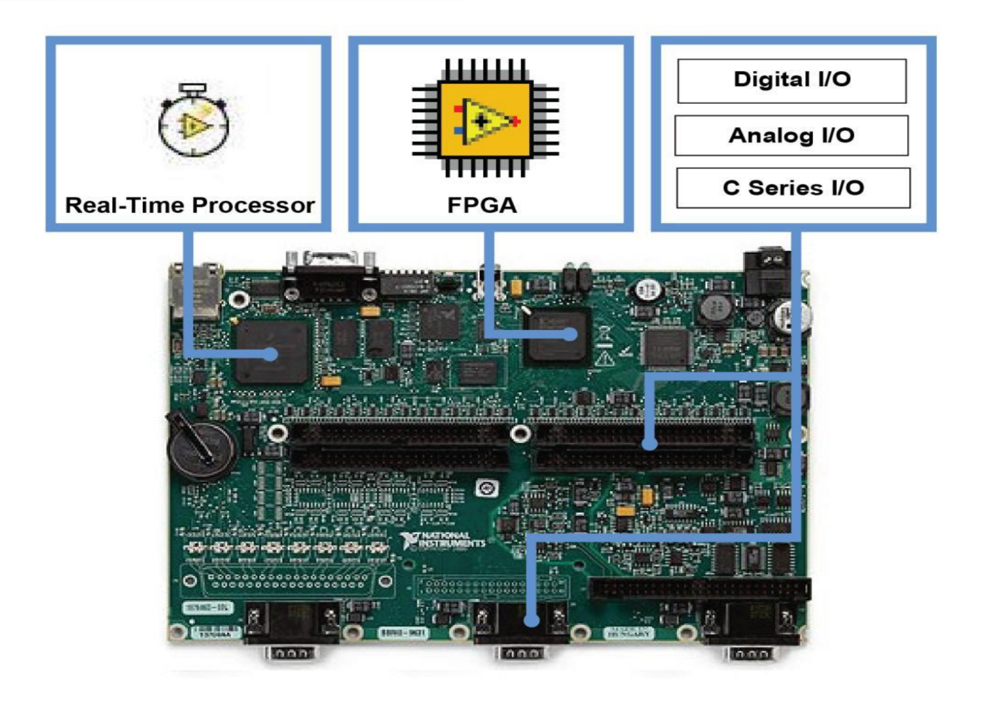
DaNi National Instrument’s Robotic Starter Kit 1.0

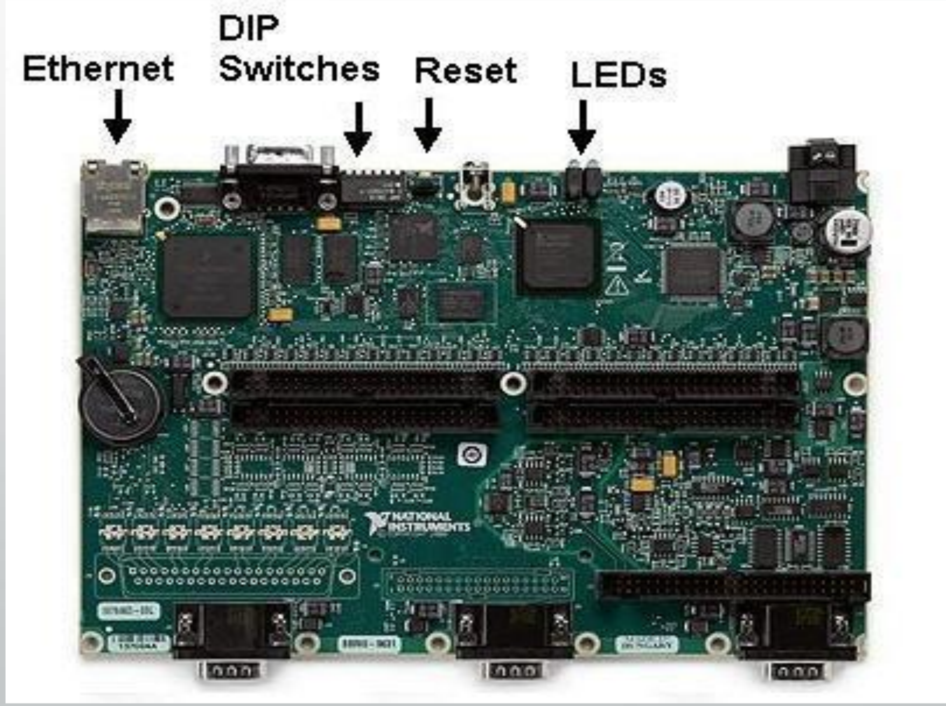
- The NI sbRIO-9632 embedded control
- Ni encoder Kit
- Servo Motors
- Ultrasonic sensor



NI sbRIO-9632 Overview

- real-time processor
- reconfigurable field-programmable gate array (FPGA)
- I/O on a single printed circuit board
- Ethernet port
- 128 MB DRAM, 256 MB non-volatile storage

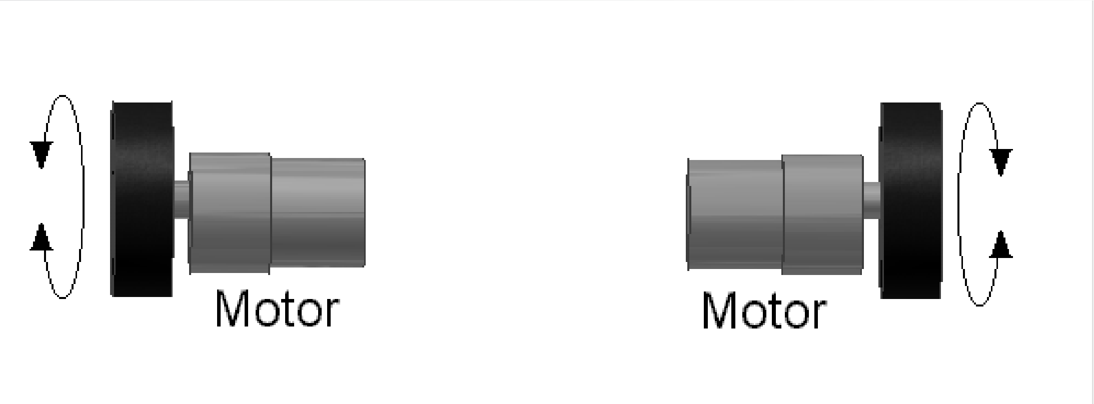




Motion Of the Robot

Controlled by two pairs of mechanically geared wheels driven by two DC servo motors

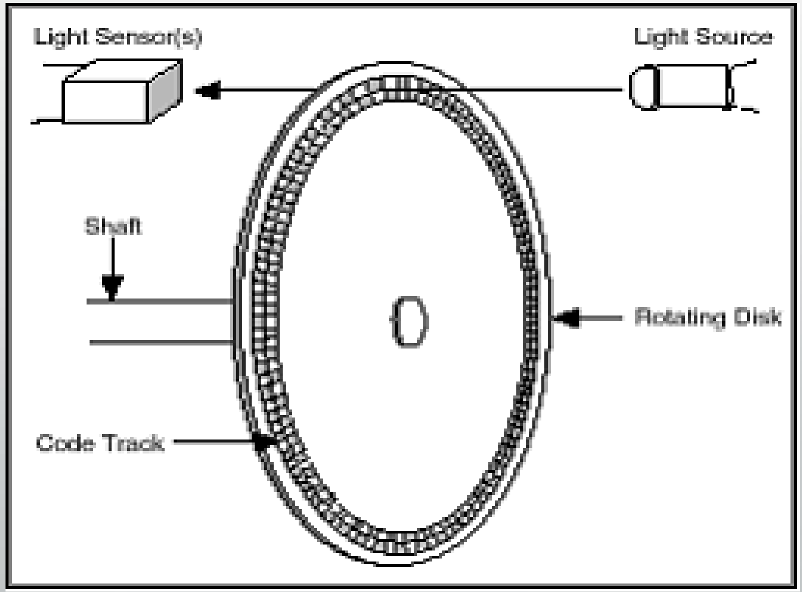
Differential steering



Encoder kit

Optical Encoders

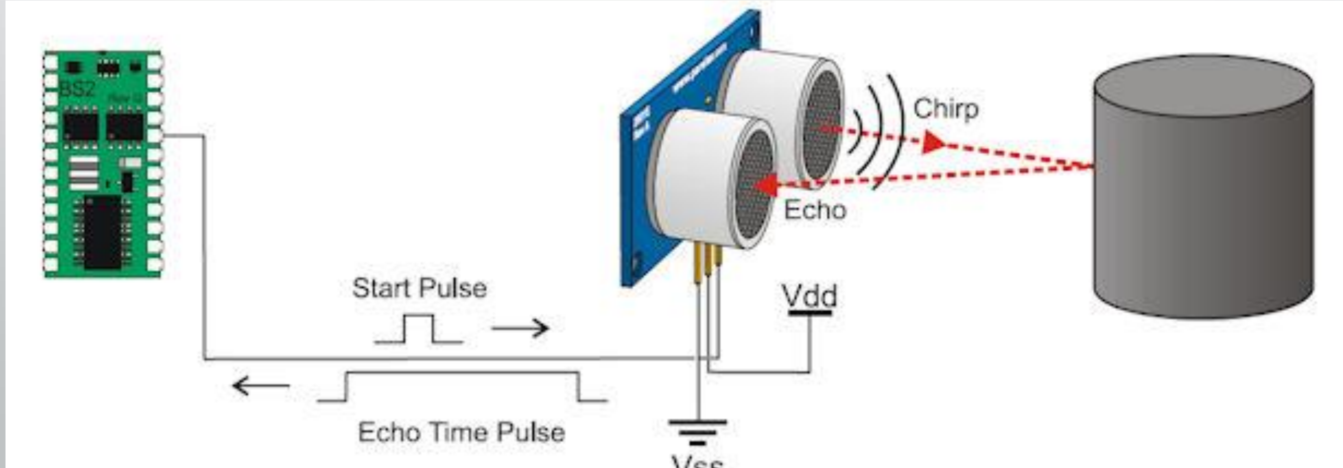
- converts the rotary displacement into digital or pulse signals
- An optical encoder uses a rotating disk, a light source, and a photodetector
- The encoder light source, pointed at a photodetector, passes through the disk sectors. As the disk rotates, these patterns interrupt the light emitted onto the photodetector, generating a digital or pulse signal output.



Ultrasonic Sensor

- Detects objects by emitting a short ultrasonic burst and then "listening" for a reflecting signal
- The transducer is connected to the sbRIO computer and the sbRIO acquires the transducer signal data
- Time-of-flight is the time that elapsed between emitting a packet of (ultrasonic) pressure waves, and receiving the reflection
- The time is measured at the FPGA in ticks of the FPGA's 40 MHz clock, meaning each 1 tick is equal to 25 nanoseconds.
- The FPGA converts ticks to time-of-flight data and then to distance in meters

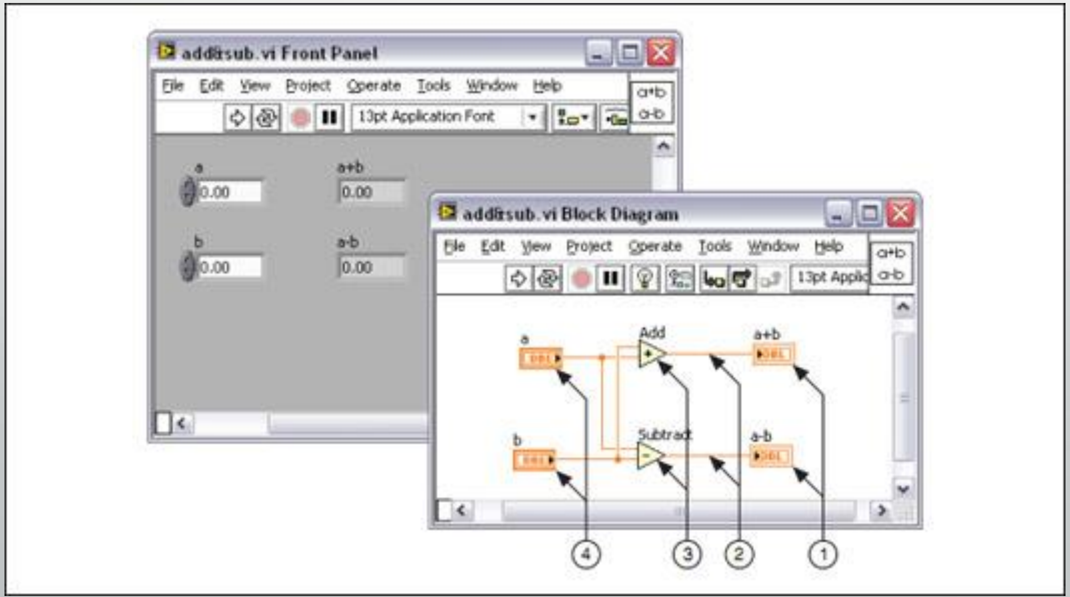
d = c * t / 2



LabView

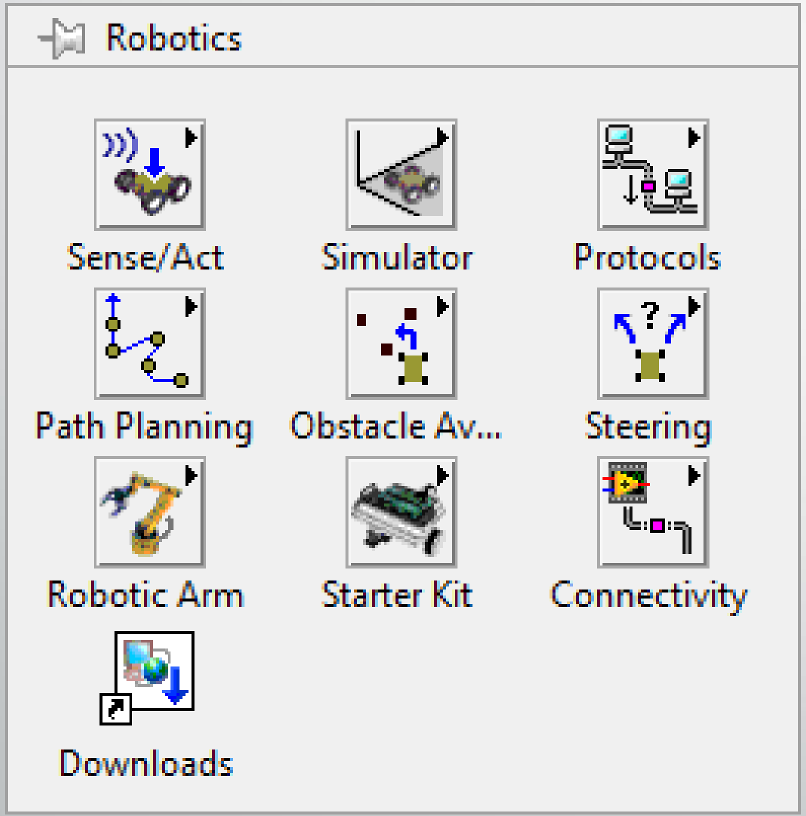
- A system-design platform and development environment for a visual programming language
- LabVIEW programs are called virtual instruments, or VIs, because their appearance and operation often imitate physical instruments
- **Front Panel**: The user interaface for the VI

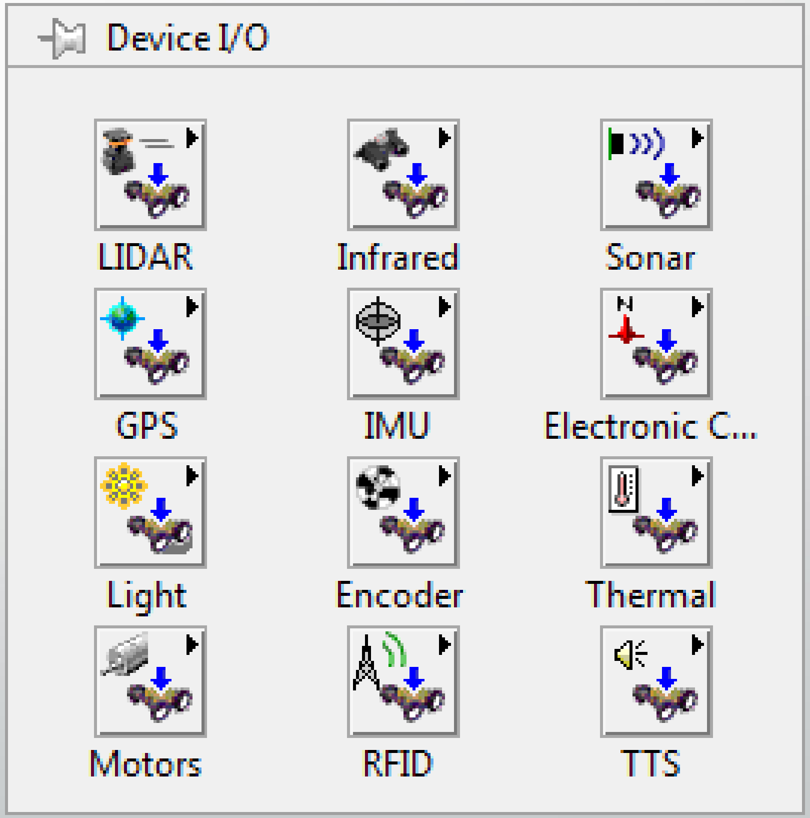
  The Controls palette contains the controls and indicators we use to create the front panel.
- **Block Diagram**: includes terminals, subVIs, functions, constants, structures, and wires, which transfer data among other block diagram objects



LabVIEW Robotics Module

- The LabView Robotics Module includes all of the software tools needed to design a sophisticated autonomous or semi-autonomous system. LabVIEW Robotics also includes a new set of VIs to configure, control, and retrieve data from the most commonly used sensors.





Programming DaNi

The Algorithm consists mainly from 4 parts

1. Initialization and Obstacle Detection based Ultrasonic Transducer
2. Motion control, based on driving the motors with appropriate velocities
3. obstacle avoidance, based on the Vector Field Histogram method
4. Localization based on Dead Reckoning with Optical Encoders

Initialization

- Initialize Starter kit 1.0

  Begins a communication session with the FPGA on the robot

- Initialize Steering Frame

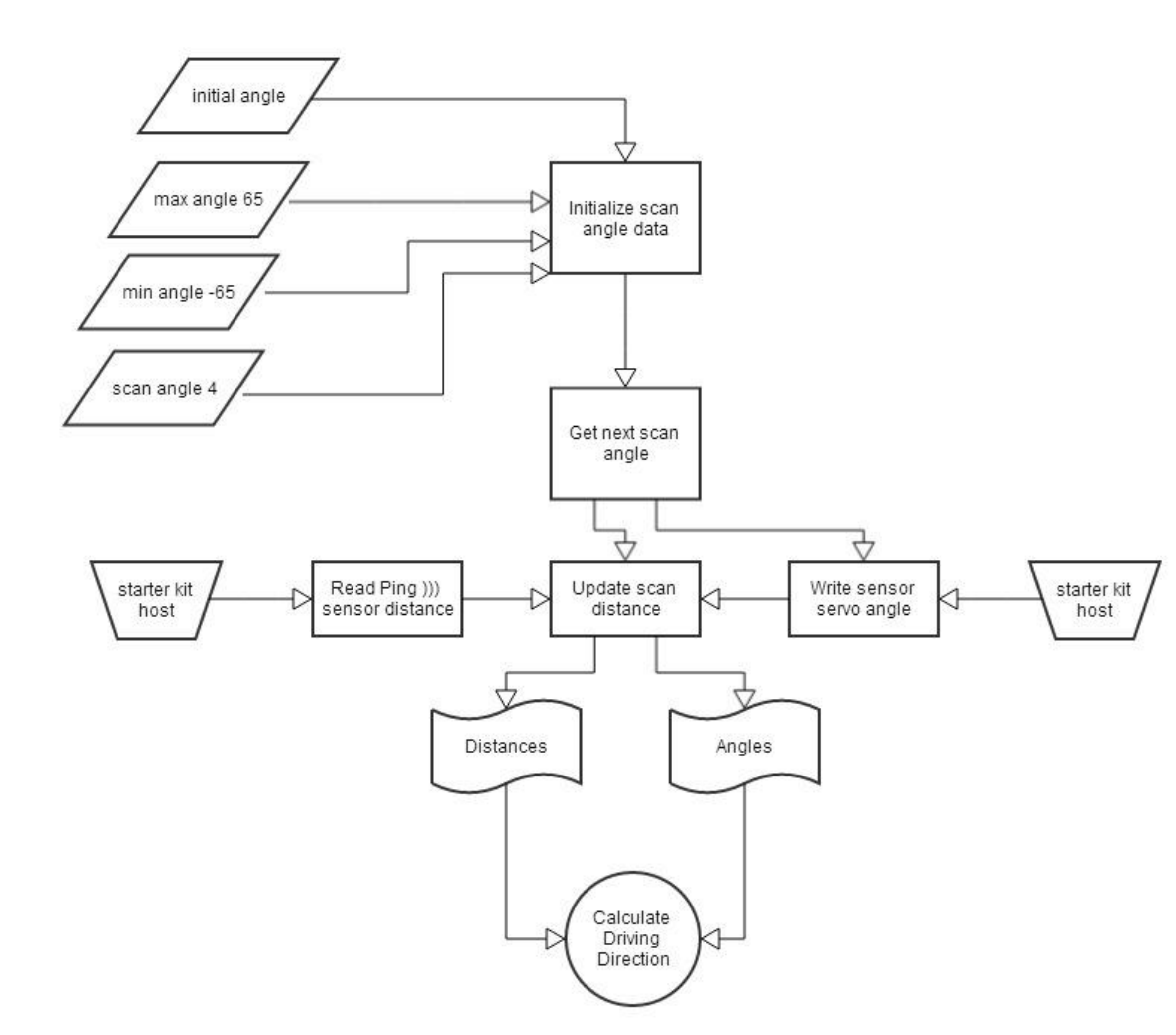
  We can use the steering frame object with the Steering VIs to implement steering for the Starter Kit robot.

- Intitialize Scan Angle Data

  The Initialize VI creates a data structure for the angle and distance information

Calculate Driving Direction

- Get Next Scan Angle
- Write Sensor Servo Angle
- Read Ping))) Sensor Distance
- Update Scan Distance
- Calculate Driving Direction

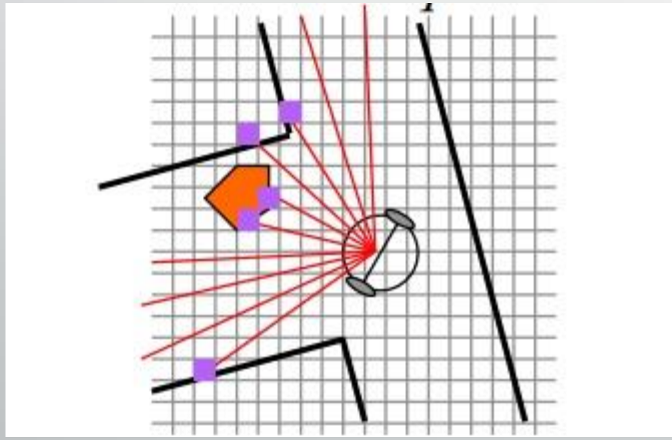


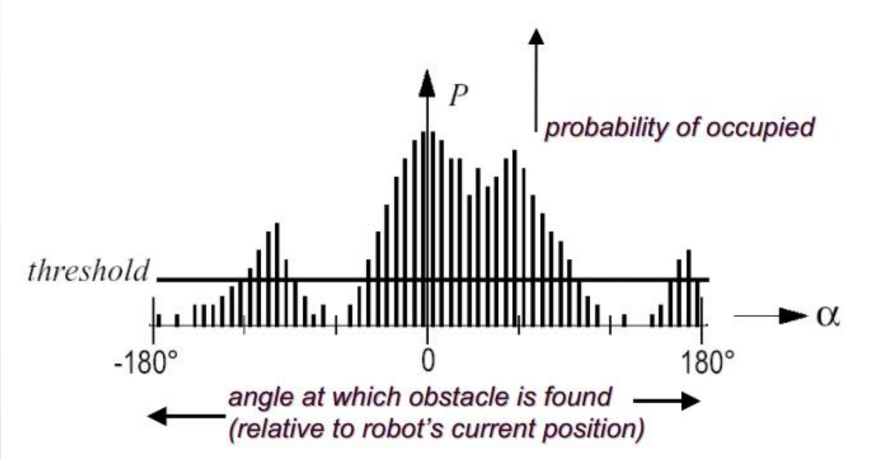
Obstacle Avoidance with VFH

Steps of the algorithm

- The Certainty Grid is updated
- A moving window centered around the robot is generated
- A polar histogram of the obstacle densities of the moving window is created

  Each sector in the polar histogram contains a value representing the polar obstacle density in that direction. In that stage, the algorithm selects the most suitable sector from among all polar histogram sectors with a low polar obstacle density, and the steering of the robot is aligned with that direction.

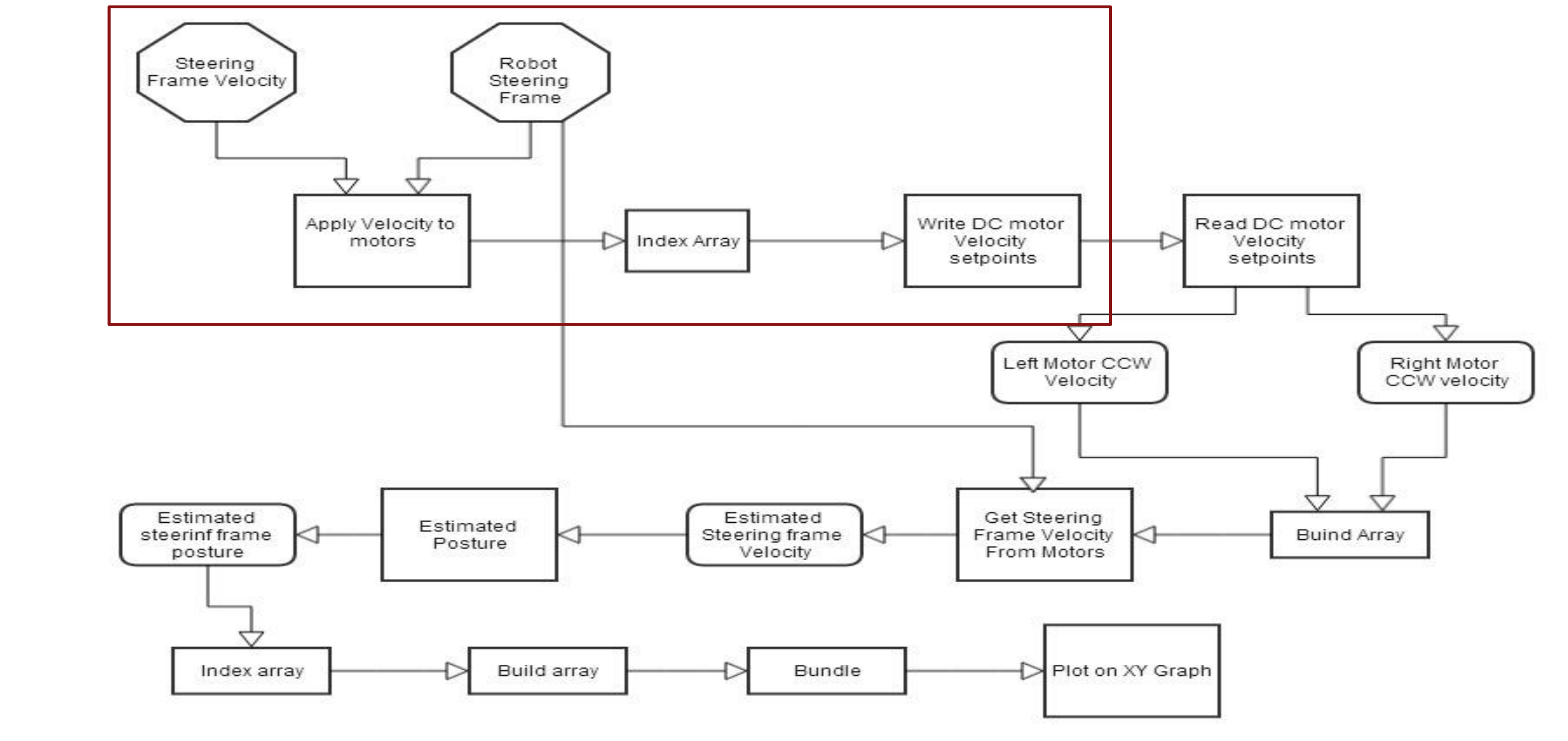




Motion control

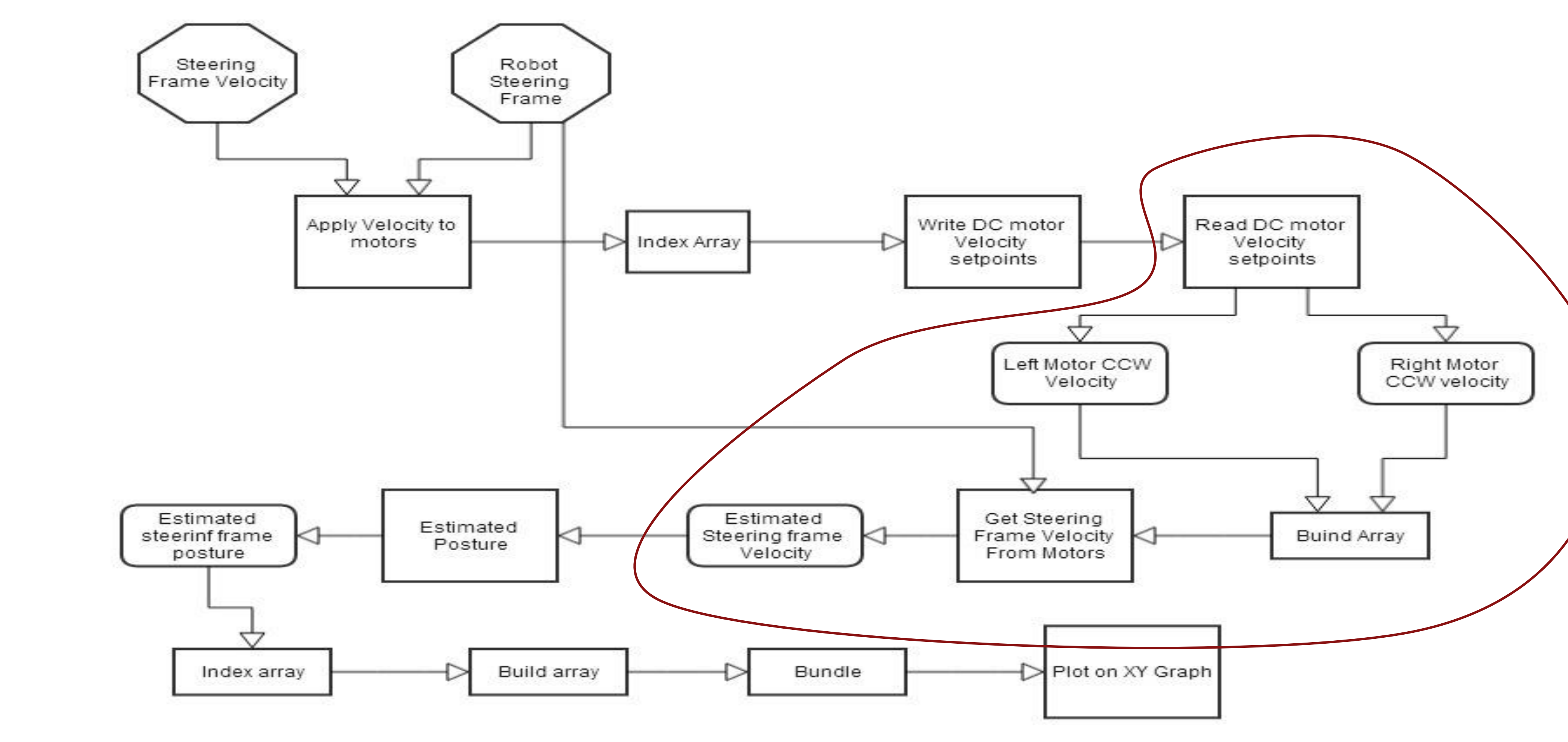
The movement of the NI Robotic Starter Kit 1.0 is performed through defining CCW (counter clock wise) set point velocity to the Servo Motors.

- Apply velocity to motors
- Write DC motor Velocity Setpoints



Localization

- Read DC Motor Velocity
- Get Velocity from Motors



Dead Reckoning

Is the process of calculating one's current position by using a previously determined position.In robotics, the necessary information is often obtained by measuring wheel revolutions, with optical encoders

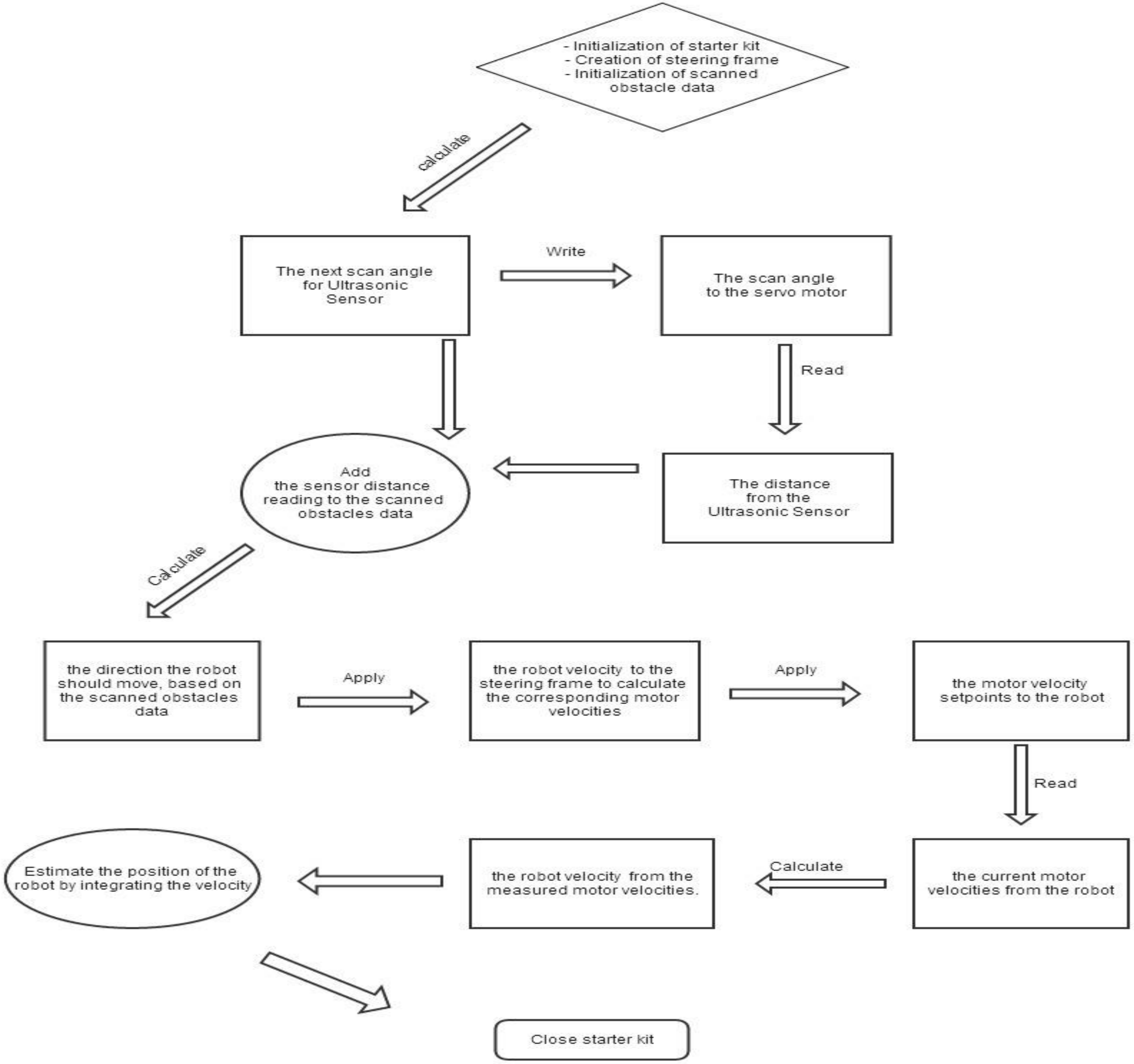
Optical encoders acquire data of the rotation of the robot wheels and allow the deduction of the distance covered by the robot in the x and y directions, as well as the change in the orientation of the robot.

- The optical encoders on the Pitsco DC motors require a 5V supply and produce 100 counts/revolution (CPR) and 400 pulses/revolution (PPR).
- 0.3192 m/revolution / 100 counts/revolution = 0.003192 m/count
- 0.3192 m/revolution / 400 pulses/revolution = 0.000798 m/pulse
- Considering the 400 pulses/revolution and DaNI’s wheel circumference the resolution in linear travel distance is 0.000798 m or 0.798 mm

Estimation of Posture

The estimation of posture is based on the estimated steering frame velocity and the initial posture of the robot. For a differential-drive robot like DaNi the position can be estimated starting from a known position by integrating the movement

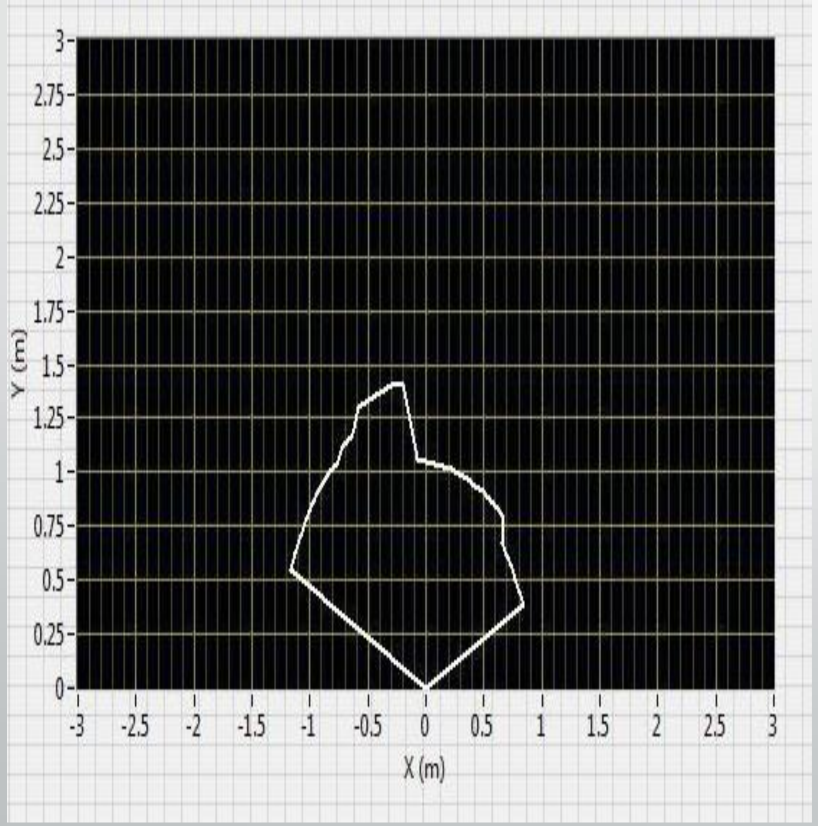
- Integrate the angular velocity component to estimate the angular position.
- Rotate the X and Y velocity components to be aligned with the global frame of reference.
- Integrate the rotated X and Y velocity components to estimate the X and Y position.

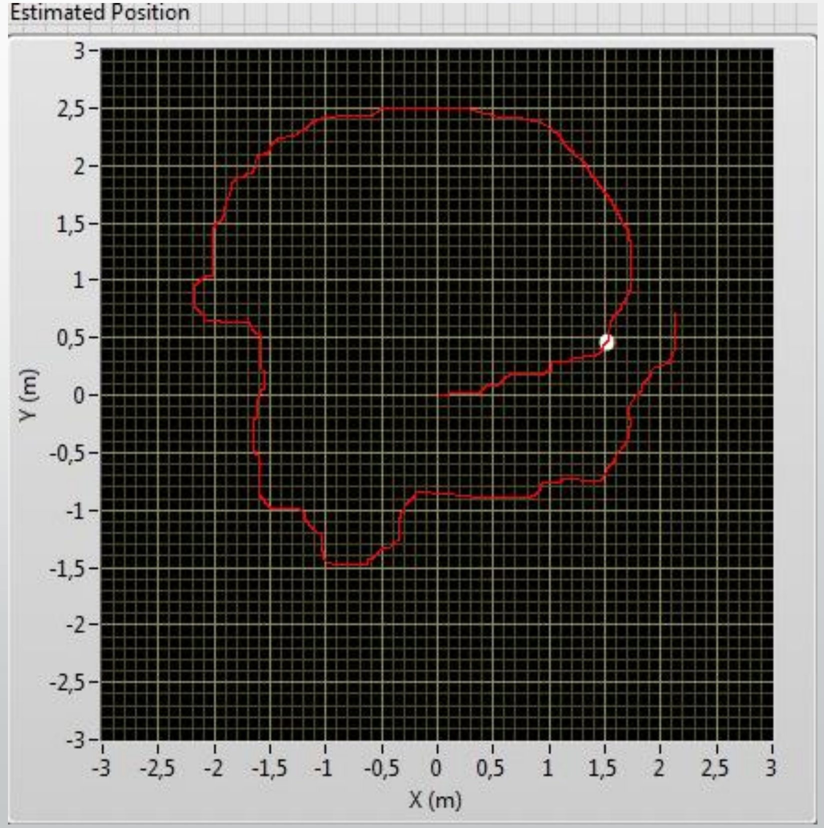

- Initialization of starter kit
- Creation of steering frame
- Initialization of scanned obstacle data
calculate
The next scan angle for Ultrasonic Sensor
Write
The scan angle to the servo motor
Read
The distance from the Ultrasonic Sensor
Add the sensor distance reading to the scanned obstacles data
Calculate
the direction the robot should move, based on the scanned obstacles data
Apply
the robot velocity to the steering frame to calculate the corresponding motor velocities
Apply
the motor velocity setpoints to the robot
Read
the current motor velocities from the robot
Calculate
the robot velocity from the measured motor velocities.
Estimate the position of the robot by integrating the velocity
Close starter kit

XY Graphs

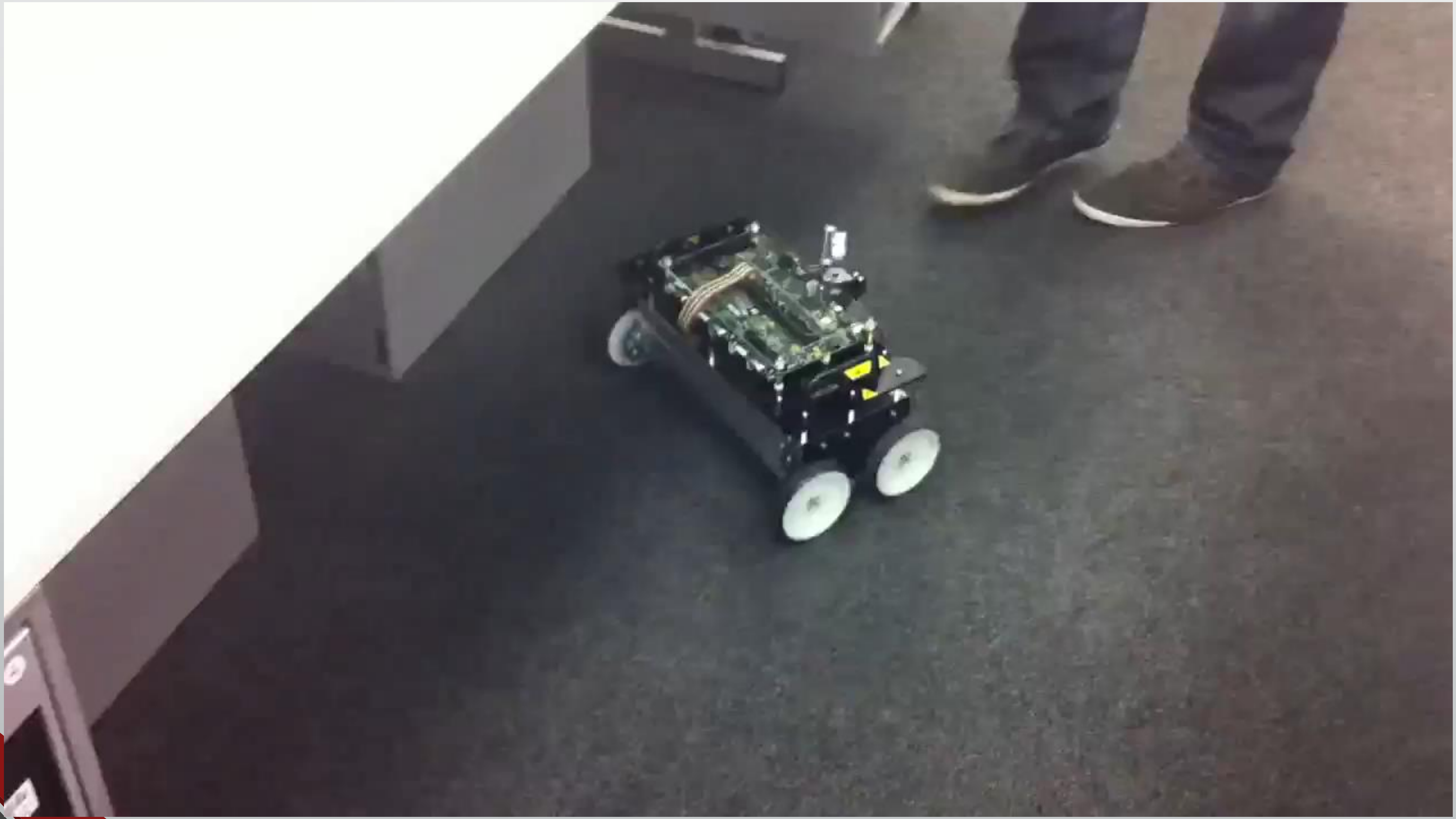
Data are transmitted back to a host computer through Ethernet connection

- The first graph shows us the measurements we have obtained from the ultrasonic sensor
- the second graph shows us the estimated position of the robot





Conclusions



Thank you verry much for your Interest