Autonomous Lawn Mower

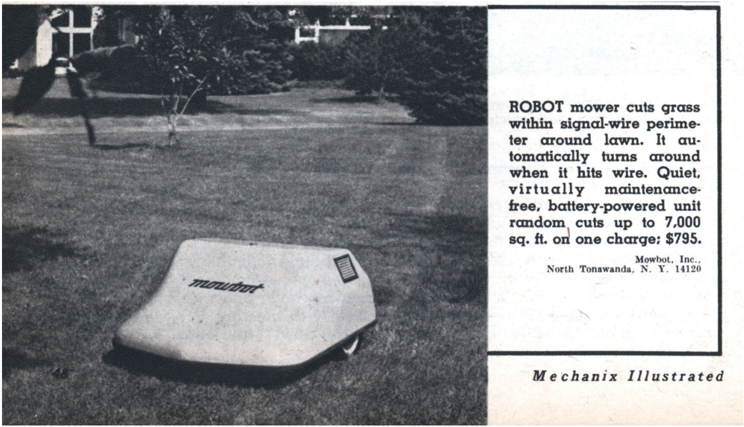
January 28, 2016

Amalie V. Petersen
Julien Bréhin
Mads R. Gotthardsen
Niels Skov Vestergaard
Romaric Destremau
Thomas Rasmussen

SICT
Aalborg University
Denmark

AALBORG UNIVERSITY
DENMARK

Agenda

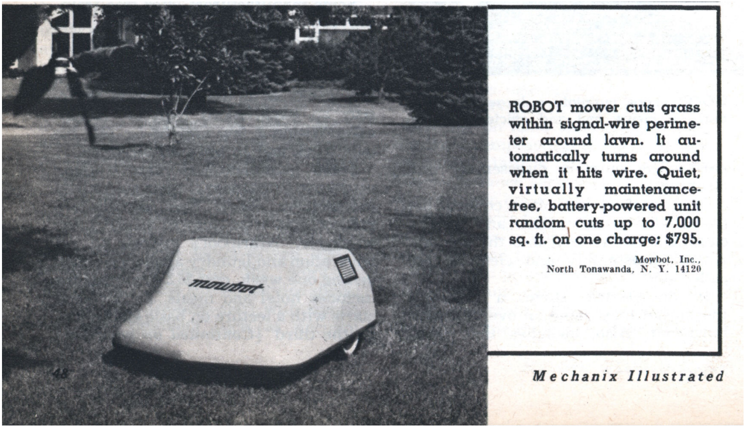


ROBOT mower cuts grass within signal-wire perimeter around lawn. It automatically turns around when it hits wire. Quiet, virtually maintenance-free, battery-powered unit random cuts up to 7,000 sq. ft. on one charge: $795.

Mowbot, Inc.,
North Tonawanda, N. Y. 14120

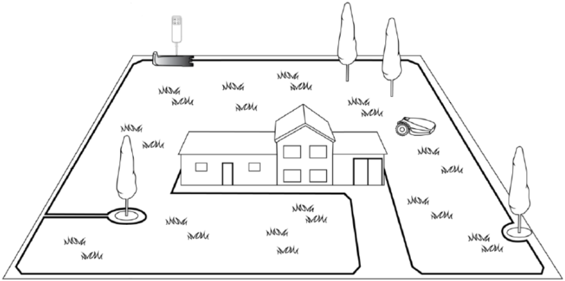
48

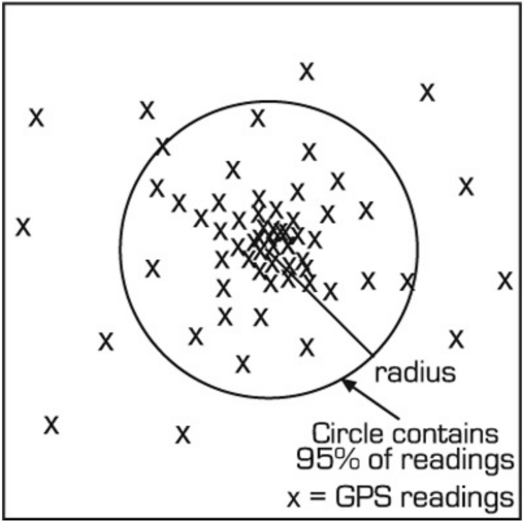
Mechanix Illustrated



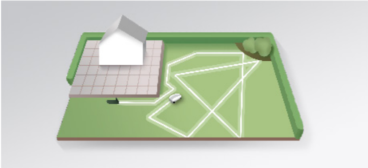
$\$795(1969) \approx \$5296(2016)$

3

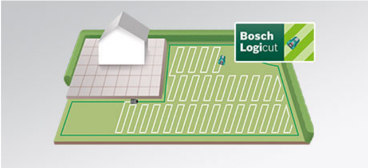



radius
Circle contains
95% of readings
x = GPS readings

Two main types:

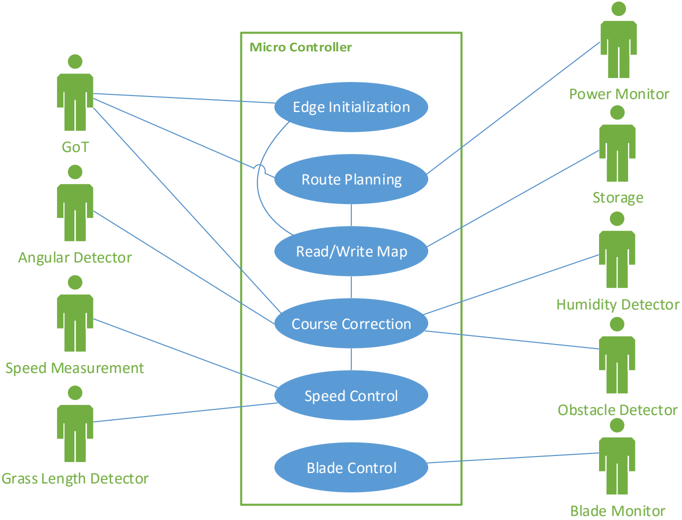


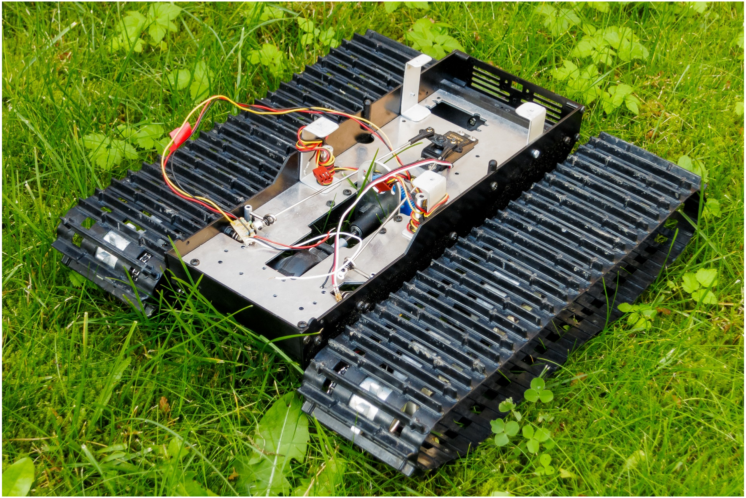
1. Random direction



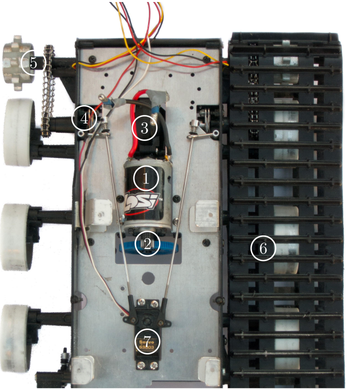
2. Parallel line

Design Considerations
Use-case Design



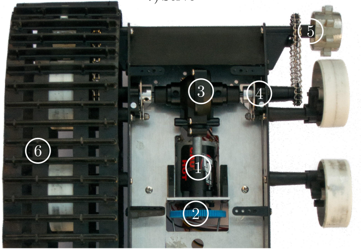


Design Considerations
Prototype Vehicle



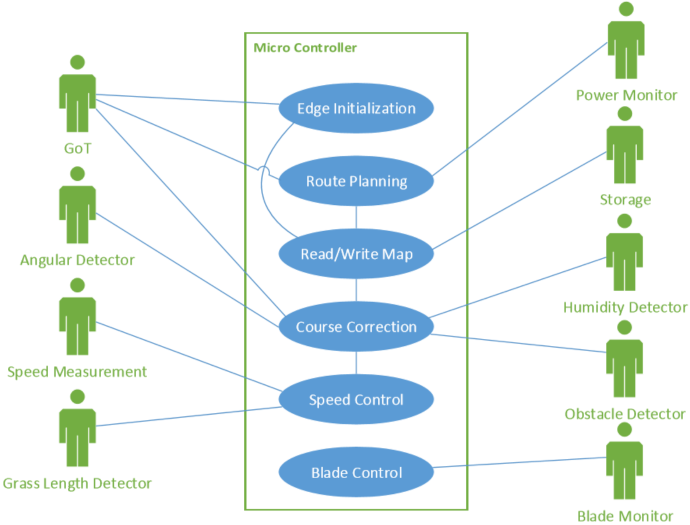
Up Side View

1) Motor
2) Motor Gear
3) Differential Gears
4) Brakes
5) Drive Wheel
6) Belts
7) Servo

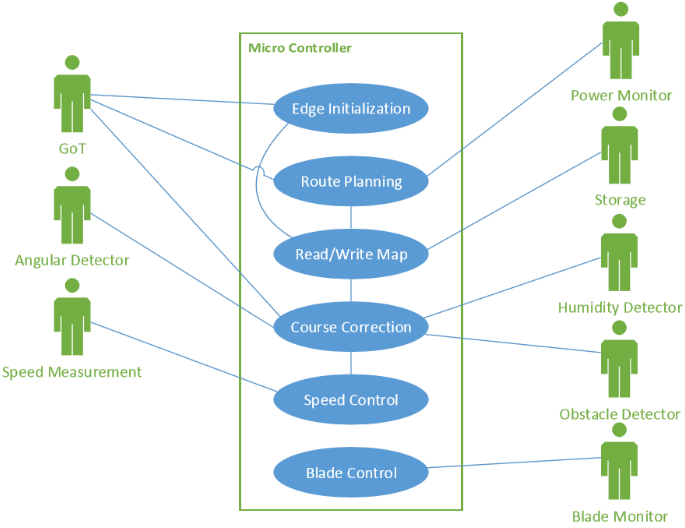


Down Side View

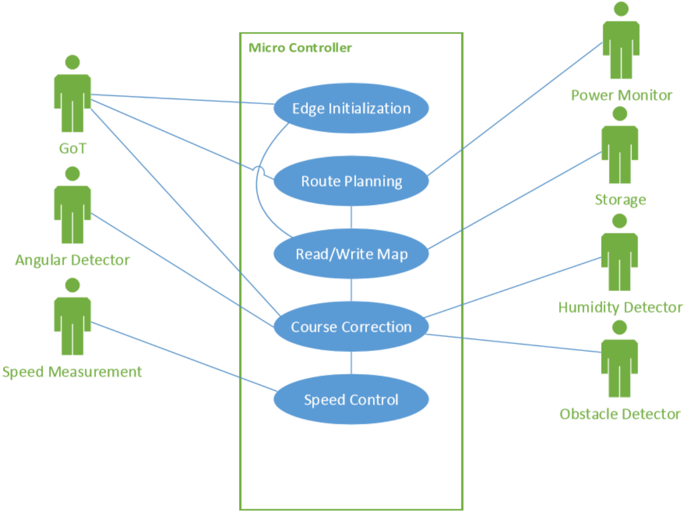
Prototype Requirements



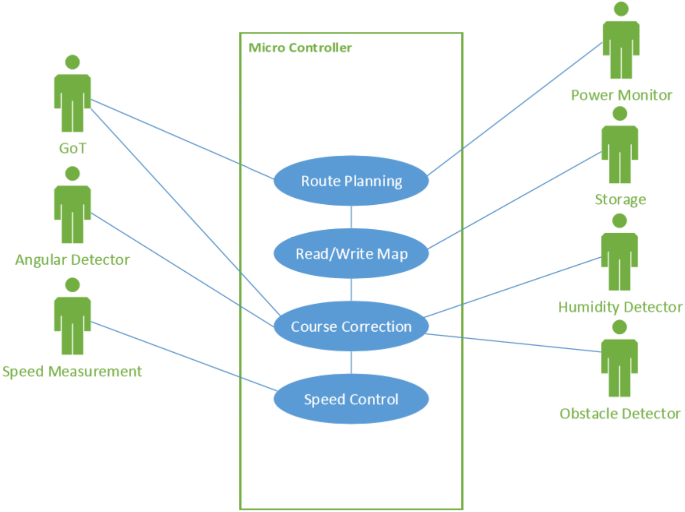
Prototype Requirements



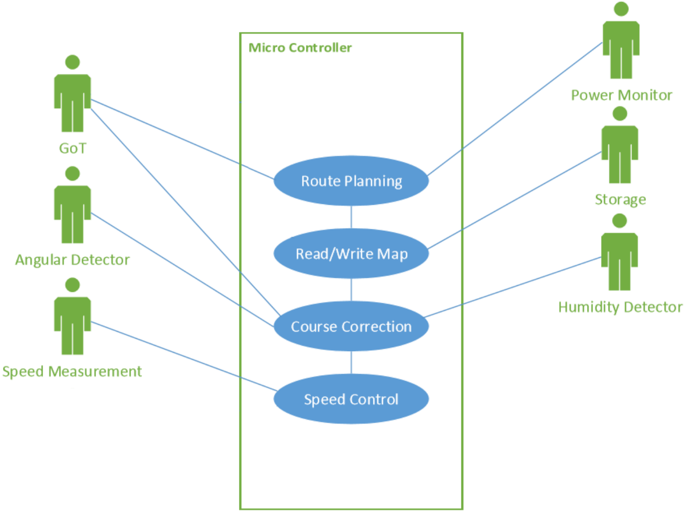
Prototype Requirements



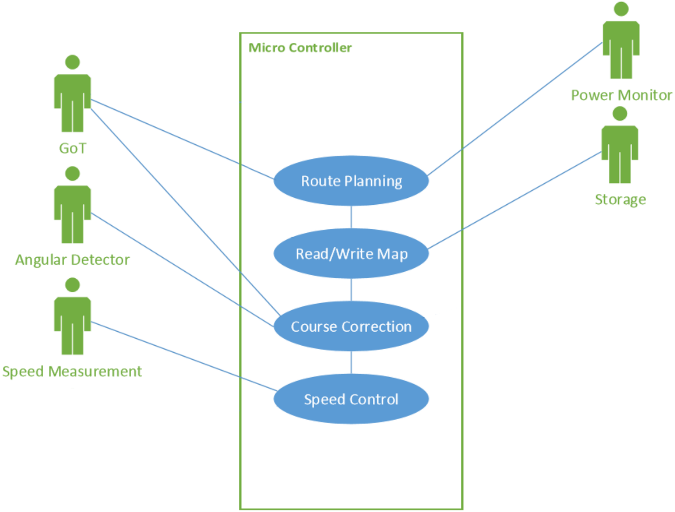
Prototype Requirements



Prototype Requirements

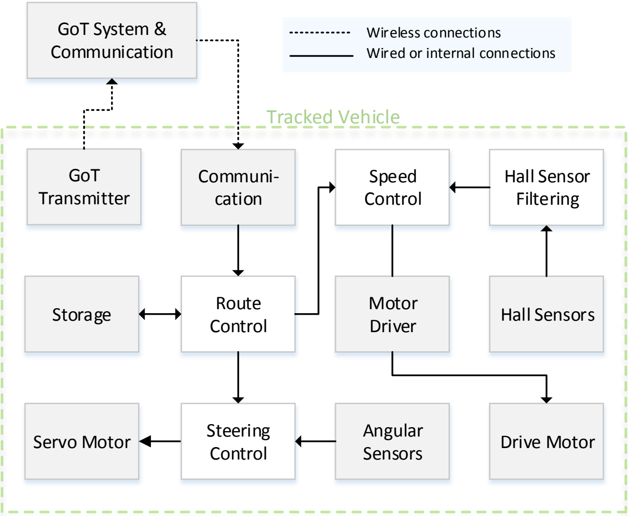


Prototype Requirements



Prototype Requirements

1. **It shall be possible for the vehicle to receive its own location wirelessly from the GoT system, through a computer.**
2. **It shall be possible for the prototype to disregard incorrect packets transmitted from the computer**
3. **The prototype must be able to disregard erroneous coordinates sent from the GoT system**
4. **The prototype must be able to access the route, which it has to follow, from a storage space located on the vehicle**
5. **The prototype must be able to shut down, if the battery voltage is below its cut-off specification**
6. **It shall be possible for the prototype to follow a predetermined route**
7. **It shall be possible for the prototype to return to the predetermined route if disturbed**
8. **The prototype shall be able to keep a velocity on 1,4 $m \cdot s^{-1}$, when going up - or downhill and when turning**

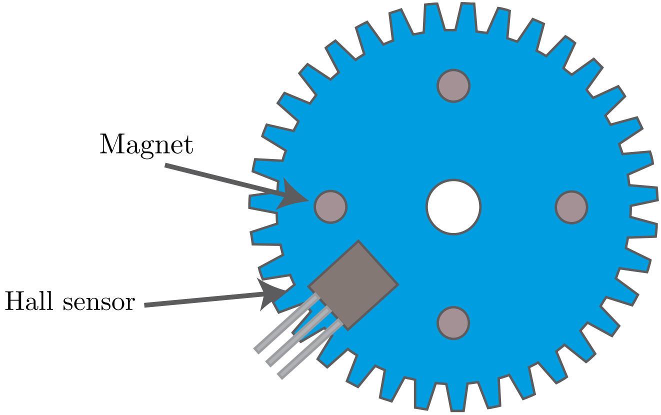

GoT System & Communication
Wireless connections
Wired or internal connections
Tracked Vehicle
GoT Transmitter
Communi-cation
Speed Control
Hall Sensor Filtering
Storage
Route Control
Motor Driver
Hall Sensors
Servo Motor
Steering Control
Angular Sensors
Drive Motor

Controller

- Arduino Mega 2560

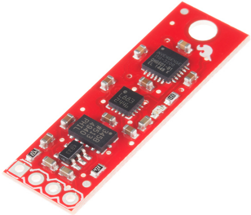
Velocity Sensor

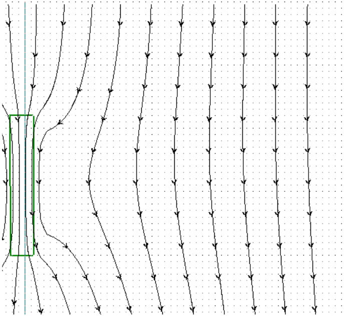
- Hall Sensor



Angular Sensor

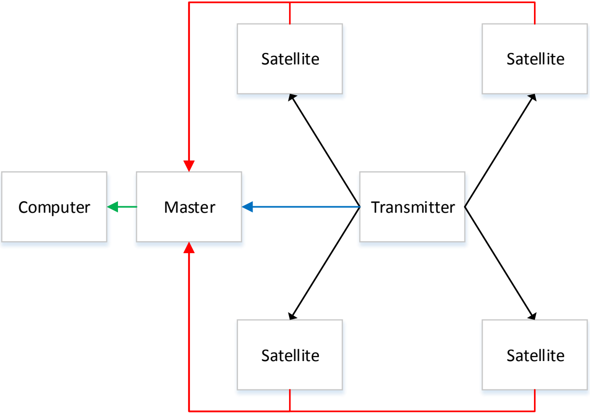
- HMC5883L Magnetometer





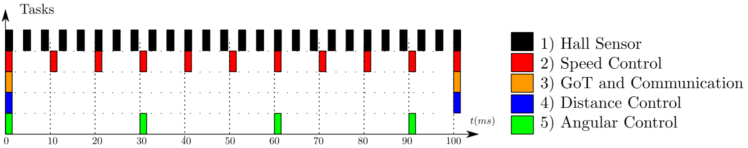
Position Sensor

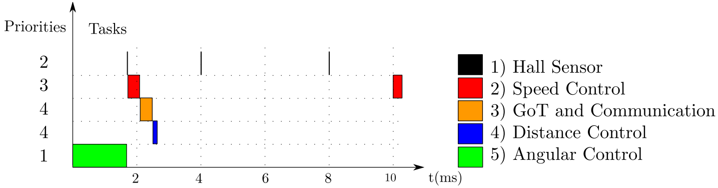
- Games on Track system (GoT)



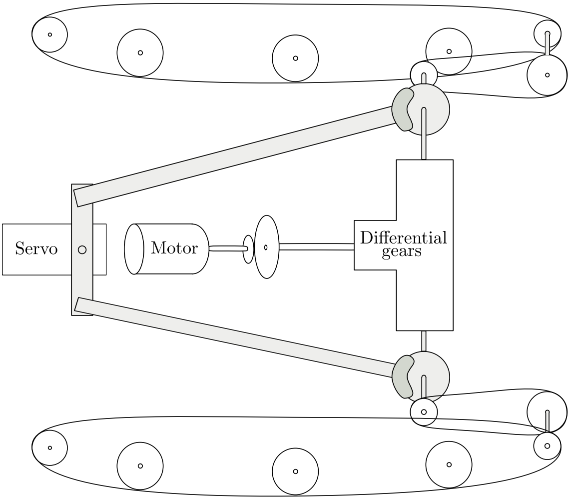
- Communication
- Storage
- Motor driver
- Battery and Power monitor

- Real Time Operating System
  - KRNL by Jens Dalsgaard Nielsen

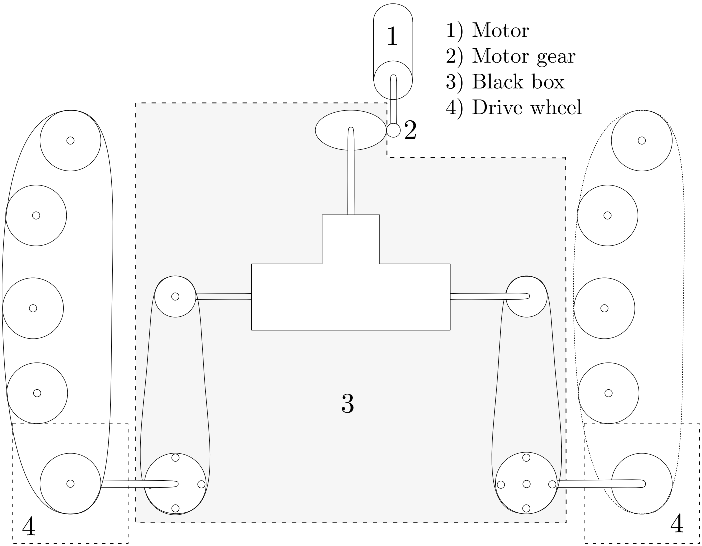




Modeling of the Vehicle

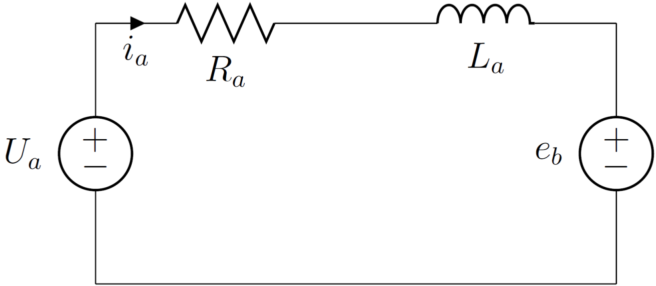


Velocity Model

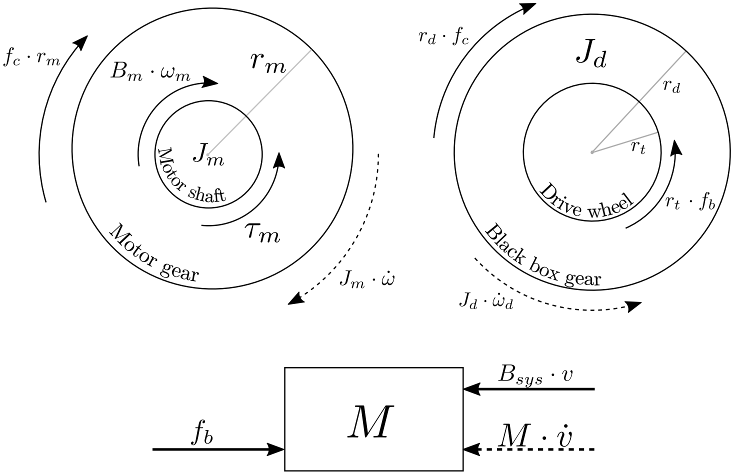


Velocity Model

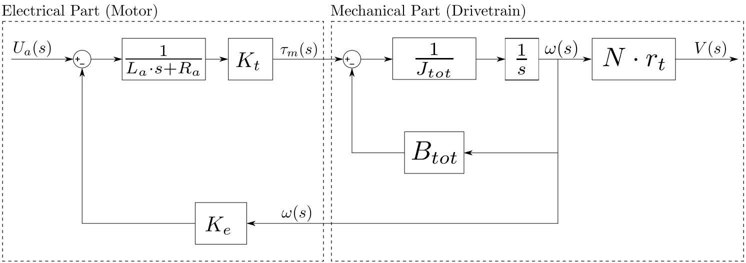
Electrical diagram

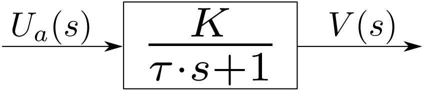


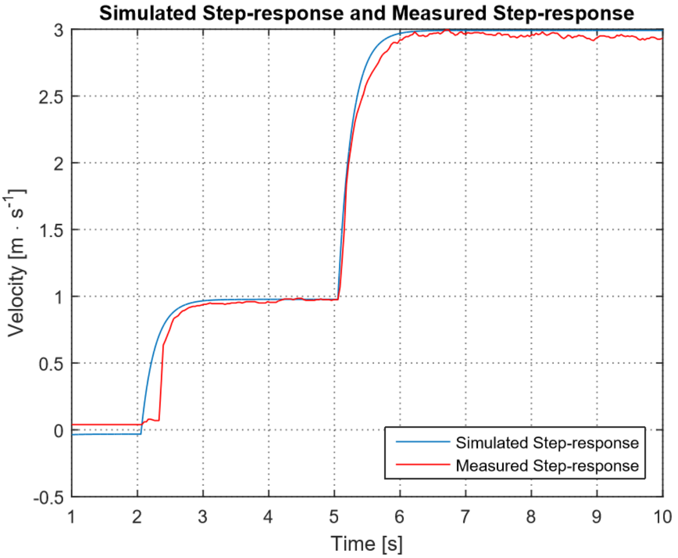
$$\tau_m = K_t \cdot \frac{U_a - K_e \cdot \omega_m}{L_a \cdot s + R_a}$$


$f_c \cdot r_m$
$B_m \cdot \omega_m$
r_m
J_m
Motor shaft
τ_m
Motor gear
$J_m \cdot \dot{\omega}$
$r_d \cdot f_c$
J_d
r_d
r_t
Drive wheel
$r_t \cdot f_b$
Black box gear
$J_d \cdot \dot{\omega}_d$
$B_{sys} \cdot v$
M
f_b
$M \cdot \dot{v}$

Velocity Model
Final Velocity Model

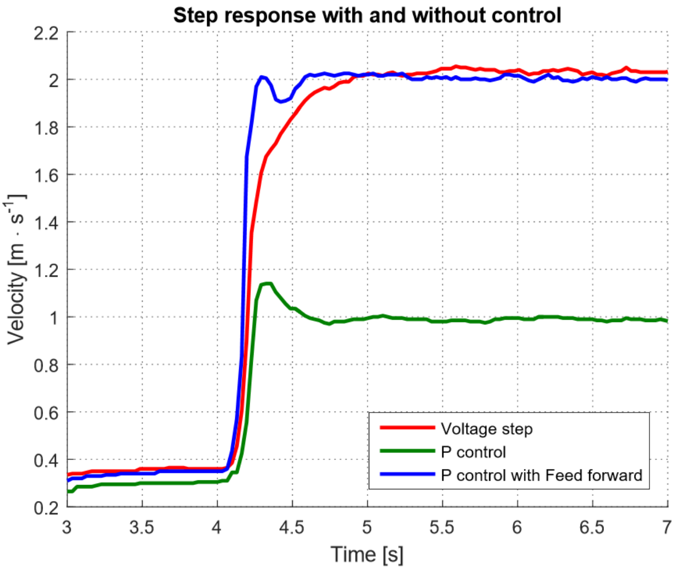





Simulated Step-response and Measured Step-response
Velocity [m · s^-1]
Time [s]
3
2.5
2
1.5
1
0.5
0
-0.5
1
2
3
4
5
6
7
8
9
10
Simulated Step-response
Measured Step-response

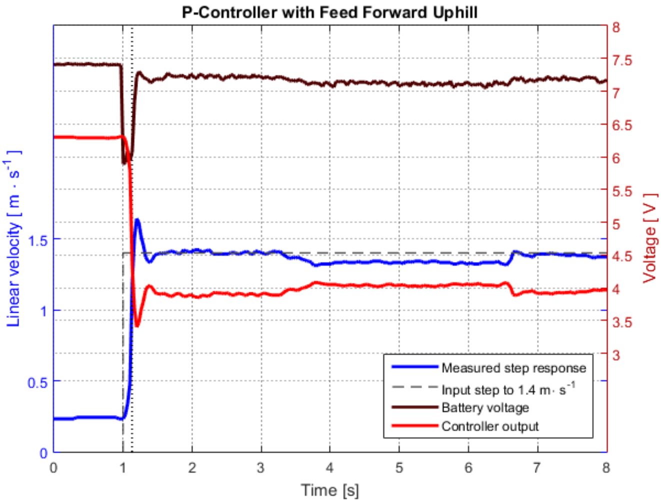
Velocity Controller

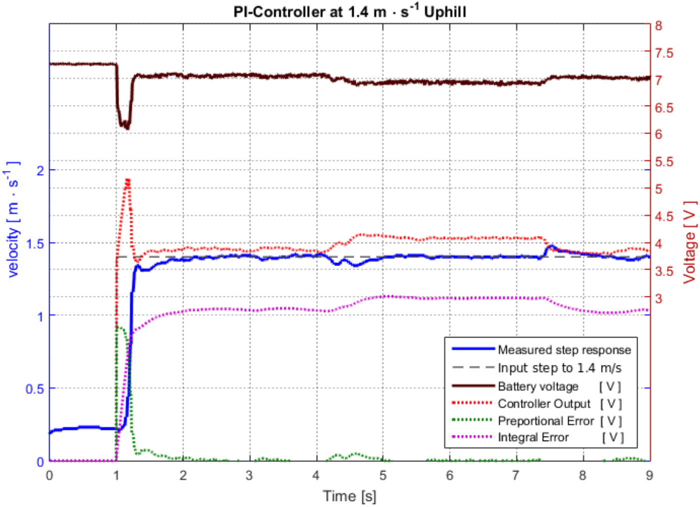
P-Controller and P-Controller with feed forward



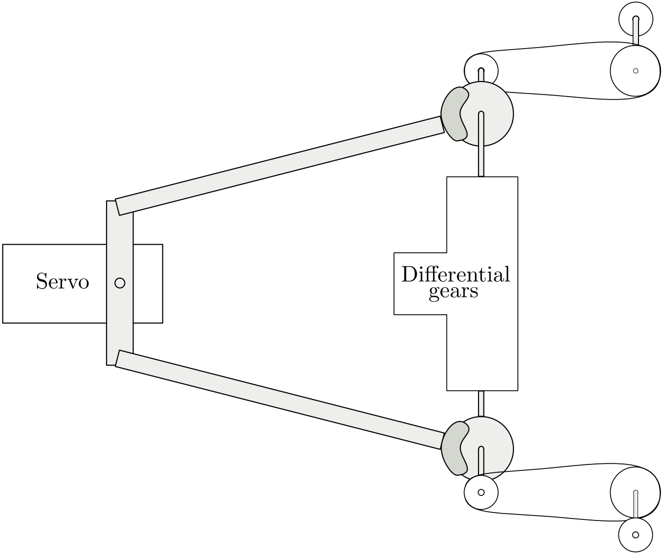
Velocity Controller

P-Controller with Feed Forward

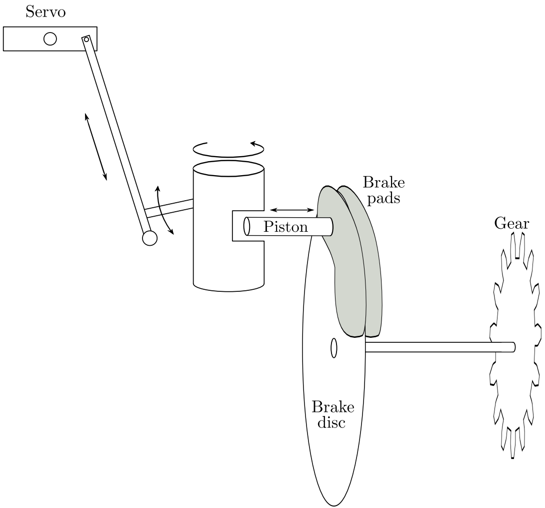




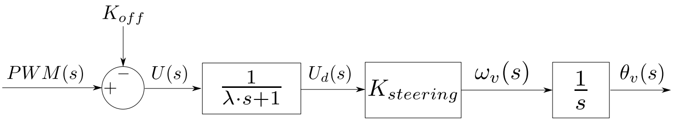
PI-Controller at 1.4 m · s^{-1} Uphill
velocity [m · s^{-1}]
Voltage [V]
Time [s]
0
0.5
1
1.5
2
3
3.5
4
4.5
5
5.5
6
6.5
7
7.5
8
0 1 2 3 4 5 6 7 8 9
Measured step response
Input step to 1.4 m/s
Battery voltage [V]
Controller Output [V]
Proportional Error [V]
Integral Error [V]

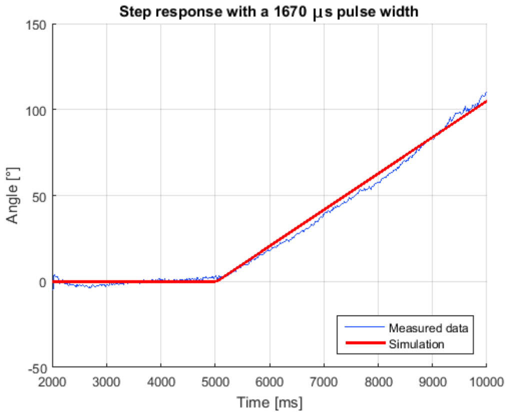


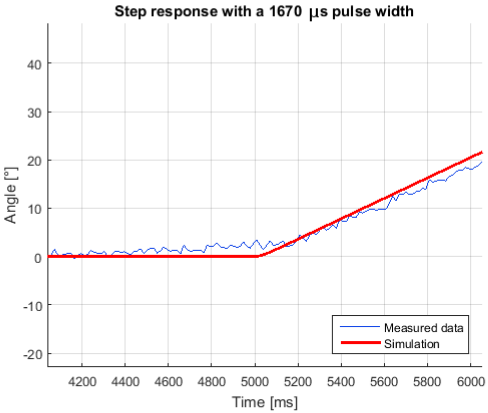
Servo
Differential gears


Servo
Brake
pads
Gear
Piston
Brake
disc

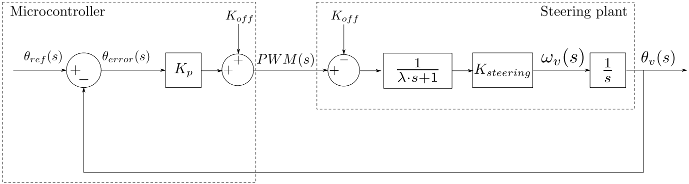
Steering Model and Control
Directional Model Enhancement




Step response with a 1670 µs pulse width
Angle [°]
Time [ms]
150
100
50
0
-50
2000
3000
4000
5000
6000
7000
8000
9000
10000
Measured data
Simulation



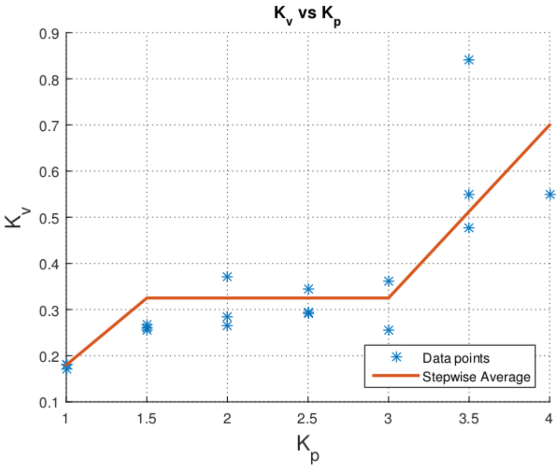
Step response with a 1670 µs pulse width
Angle [°]
Time [ms]
Measured data
Simulation



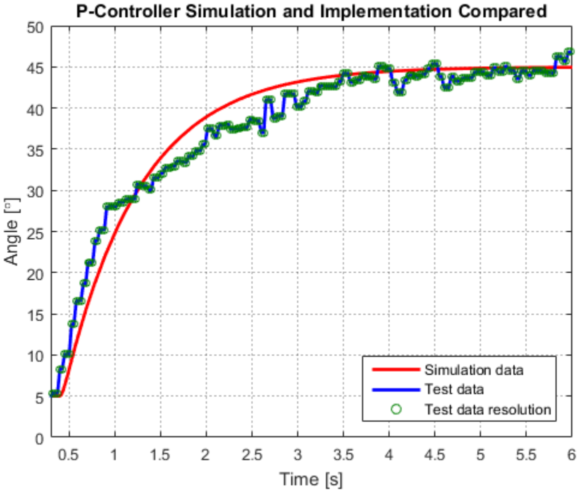
Microcontroller
Steering plant
K_{off}
K_{off}
$\theta_{ref}(s)$
$\theta_{error}(s)$
K_p
$PWM(s)$
$\frac{1}{\lambda \cdot s+1}$
$K_{steering}$
$\omega_v(s)$
$\frac{1}{s}$
$\theta_v(s)$

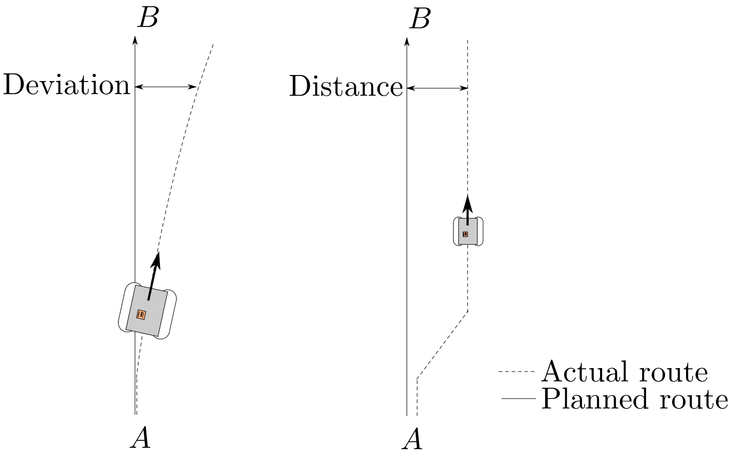
Steering Model and Control

Directional Controller (P-Controller)

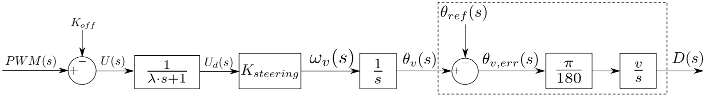


Steering Model and Control
Directional Controller Verification

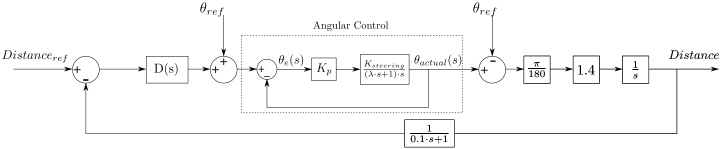


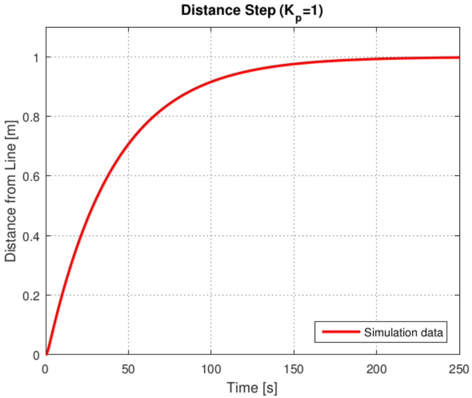

B
Deviation
A
B
Distance
A
Actual route
Planned route

$$D(s) = \frac{\pi}{180} \cdot v \cdot \frac{1}{s} \cdot (\theta_v(s) - \theta_{ref}(s))$$

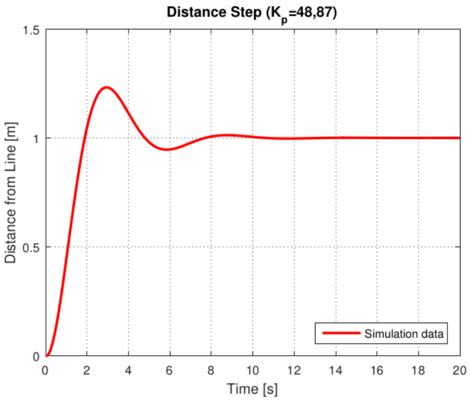


Steering Model and Control
Complete Control





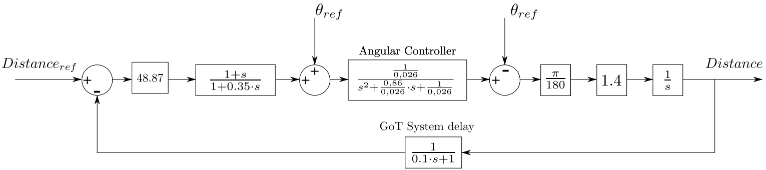
Distance Step (K_p=1)
Distance from Line [m]
Time [s]
0
0.2
0.4
0.6
0.8
1
0
50
100
150
200
250
Simulation data

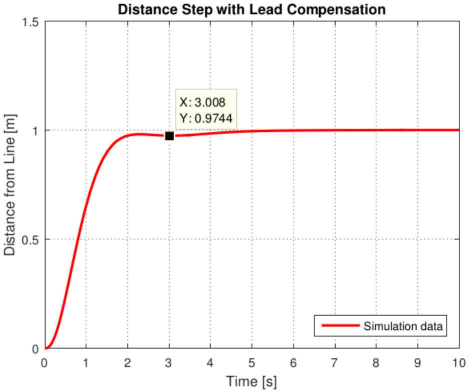


Distance Step (K_p=48,87)
Distance from Line [m]
Time [s]
Simulation data
0
0.5
1
1.5
0
2
4
6
8
10
12
14
16
18
20

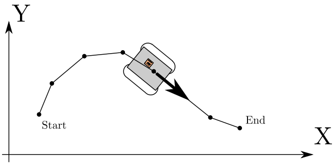
Steering Model and Control

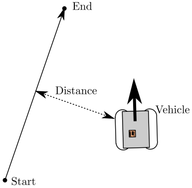
Distance Control (Lead Compensator)



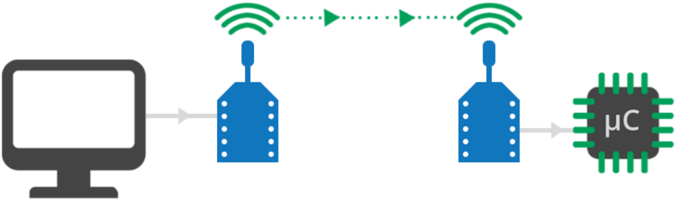


Distance Step with Lead Compensation
Distance from Line [m]
Time [s]
X: 3.008
Y: 0.9744
Simulation data
0
0.5
1
1.5
0
1
2
3
4
5
6
7
8
9
10

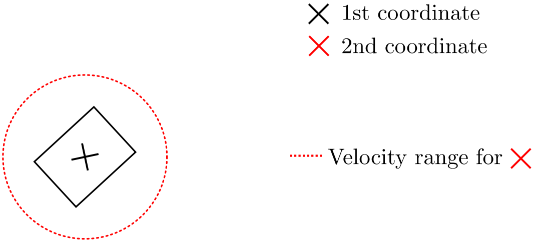

Y
X
Start
End


End
Distance
Vehicle
Start

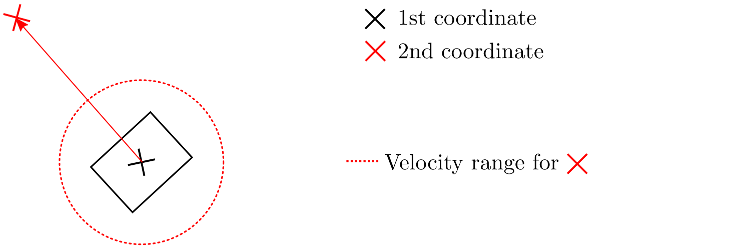
Communication



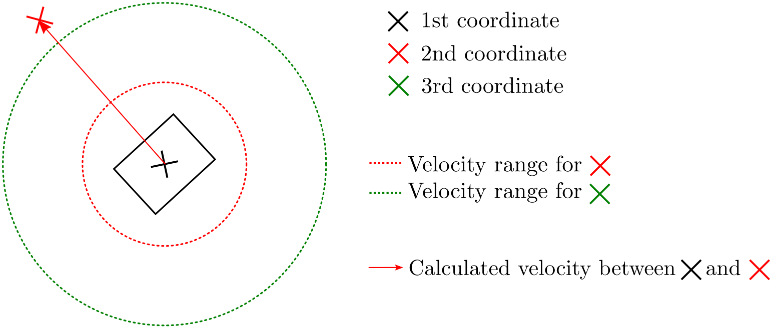
- Spikes from GoT system
- Removing unlikely data points prior transmission



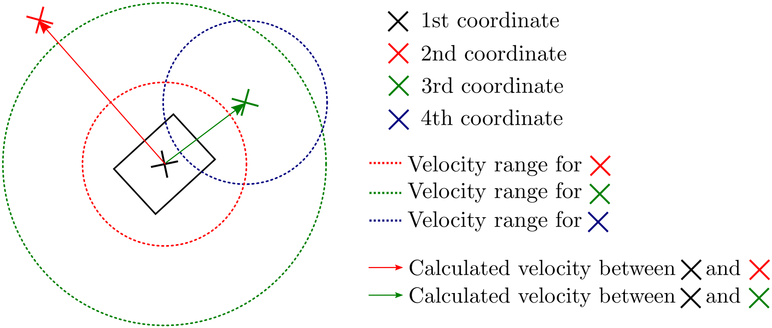
- Spikes from GoT system
- Removing unlikely data points prior transmission



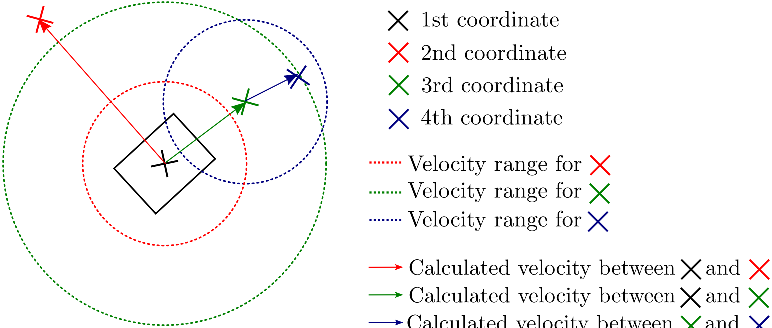
- Spikes from GoT system
- Removing unlikely data points prior transmission



- Spikes from GoT system
- Removing unlikely data points prior transmission



- Spikes from GoT system
- Removing unlikely data points prior transmission



Communication
Basis

- XBee modules

- Data path:
  GoT computer $\rightarrow$ XBee $\rightarrow$ XBee $\rightarrow$ microcontroller
- No re-transmissions since data is used for feedback
- Connection less protocol
- Focus on transport layer for reliable data reception

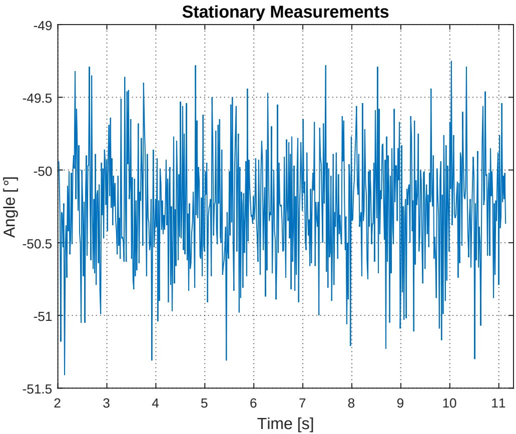
Communication
Package Transport

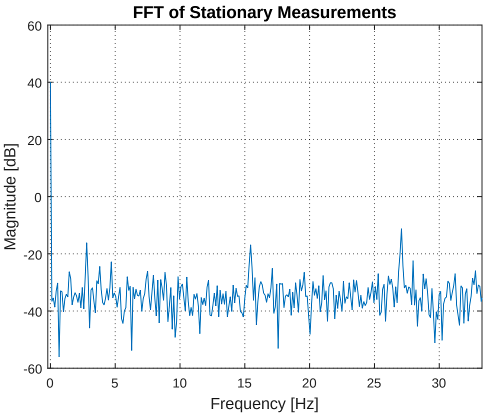
- The structure of a package

Start byte	ID	Length	Data	Checksum	End byte
Header			Body	Tail	

- ID of receiver and length of package are constant in this case
- Checksum can be larger/more effective with larger packets
- Packages may be lost
- Risk of fallacious packages being accepted is very slim

Digital Filter




FFT of Stationary Measurements
Magnitude [dB]
Frequency [Hz]
60
40
20
0
-20
-40
-60
0
5
10
15
20
25
30

- Inner loop of steering plant

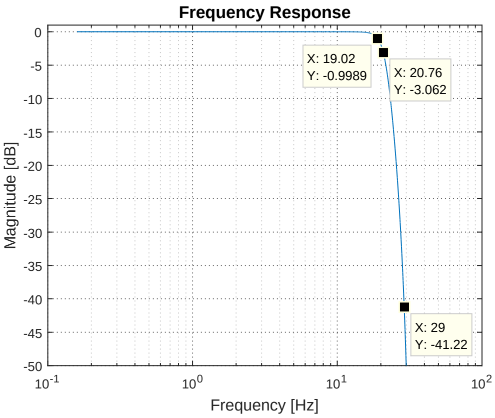
$$H(s) = \frac{K}{(0.03s + 1)s}$$

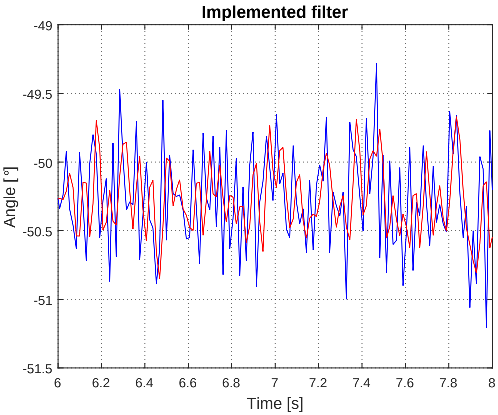
Highest placed pole is at 33,33 rad$\cdot$s^{-1}
One decade above: 333,33 rad$\cdot$s^{-1} $= 53$ Hz

- Sample rate by Nyquist

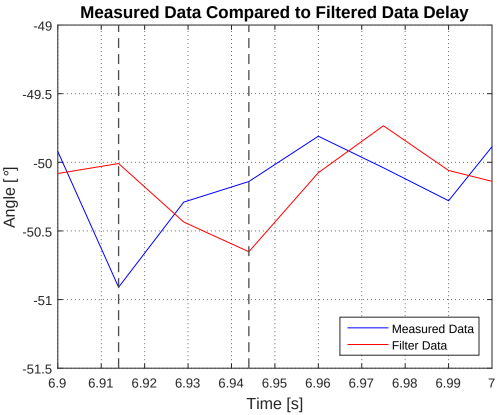
$$\Omega_s \geq 2 \cdot 53\text{ Hz} \Rightarrow \Omega_s \geq 106\text{ Hz}$$

- Max magnetometer sample rate is 75 Hz
- Filter specifications chosen iteratively
- Choice of 66.66 Hz will affect phase
- Chosen filter type: Low pass Butterworth


Frequency Response
X: 19.02
Y: -0.9989
X: 20.76
Y: -3.062
X: 29
Y: -41.22
Magnitude [dB]
Frequency [Hz]



Implemented filter
-49
-49.5
-50
-50.5
-51
-51.5
Angle [°]
6
6.2
6.4
6.6
6.8
7
7.2
7.4
7.6
7.8
8
Time [s]



Measured Data Compared to Filtered Data Delay
-49
-49.5
-50
-50.5
-51
-51.5
Angle [°]
6.9
6.91
6.92
6.93
6.94
6.95
6.96
6.97
6.98
6.99
7
Time [s]
Measured Data
Filter Data

Acceptance Test

Requirement 1

No.	Requirement	Done?
1	It shall be possible for the vehicle to receive its own location wirelessly from the GoT system, through a computer.	Yes

Acceptance Test

Requirement 1

No.	Requirement	Done?
1	It shall be possible for the vehicle to receive its own location wirelessly from the GoT system, through a computer.	Yes

GoT log file	Arduino received	Equal?
53,-23,-271	53,-23,-271	Yes
53,-24,-264	53,-24,-264	Yes
53,-24,-269	53,-24,-269	Yes
55,-24,-267	55,-24,-267	Yes
55,-23,-269	55,-23,-269	Yes
56,-23,-274	56,-23,-274	Yes
57,-22,-274	57,-22,-274	Yes
59,-21,-274	59,-21,-274	Yes
57,-22,-270	57,-22,-270	Yes
55,-23,-269	55,-23,-269	Yes

No.	Requirements	Done?
2	It shall be possible for the prototype to disregard incorrect packets transmitted from the computer.	Yes

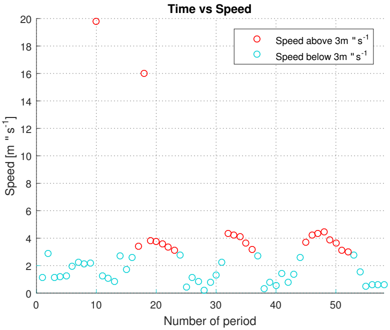
No.	Requirements	Done?
2	It shall be possible for the prototype to disregard incorrect packets transmitted from the computer.	Yes

Package	Error	Error value
1	No error	0
2	Wrong start byte	1
3	No error	0
4	Checksum fail	4
5	Wrong ID	2

Acceptance Test
Requirement 3

No.	Requirements	Done?
3	The prototype must be able to disregard erroneous coordinates sent from the GoT system.	Yes

No.	Requirements	Done?
3	The prototype must be able to disregard erroneous coordinates sent from the GoT system.	Yes



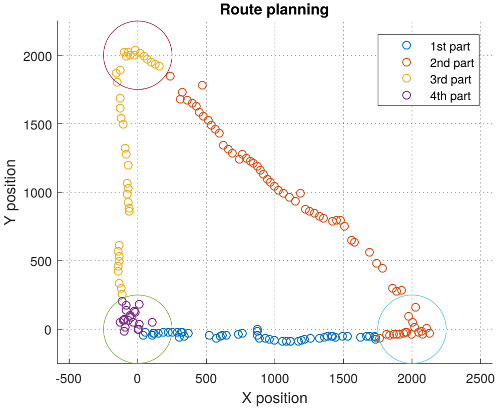
Acceptance Test
Requirement 4

No.	Requirements	Done?
4	The prototype must be able to access the route, which it has to follow, from a storage space located on the vehicle.	No

No.	Requirements	Done?
5	The prototype must be able to shut down, if the battery voltage is below its cut-off specification.	Yes

No.	Requirements	Done?
6	It shall be possible for the prototype to follow a predetermined route.	No

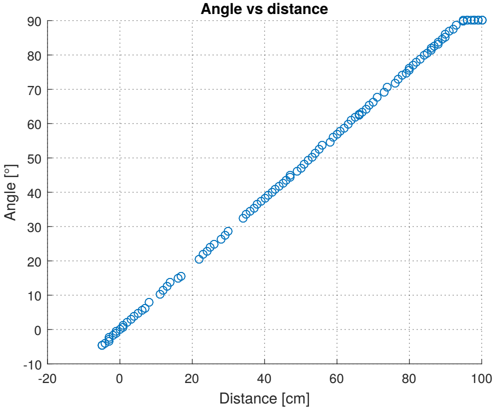
No.	Requirements	Done?
6	It shall be possible for the prototype to follow a predetermined route.	No



Acceptance Test
Requirement 7

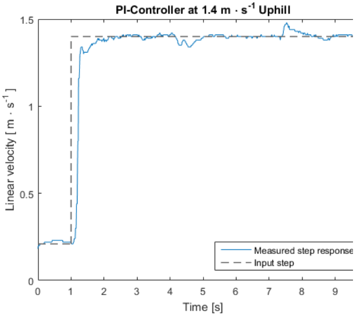
No.	Requirements	Done?
7	It shall be possible for the prototype to return to the predetermined route if disturbed.	No

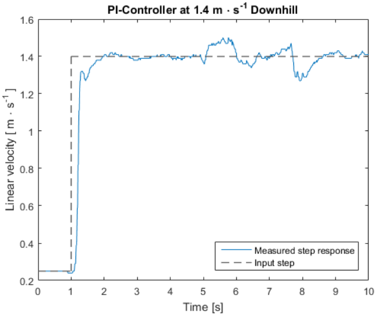
No.	Requirements	Done?
7	It shall be possible for the prototype to return to the predetermined route if disturbed.	No



No.	Requirements	Done?
8	The prototype shall be able to keep a velocity on 1,4 $m \cdot s^{-1}$, when going up or downhill and when turning.	Yes

No.	Requirements	Done?
8	The prototype shall be able to keep a velocity on 1,4 $m \cdot s^{-1}$, when going up or downhill and when turning.	Yes





Conclusion

What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements

Conclusion

What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega

Conclusion

What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling

Conclusion
What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling
- Digital filter to lower the noise

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling
- Digital filter to lower the noise
- SD card storage

Conclusion
What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling
- Digital filter to lower the noise
- SD card storage
- Mathematical models simulated

Conclusion

What has been done?

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling
- Digital filter to lower the noise
- SD card storage
- Mathematical models simulated
- Distance model tests

- Alternative navigation approaches with Games on Track system
- Belt vehicle analized to build prototype requirements
- Real Time Operating System implemented on the Arduino Mega
- XBee modules with protocol error handling
- Digital filter to lower the noise
- SD card storage
- Mathematical models simulated
- Distance model tests
- P and PI Controllers tested

Discussion
What next?

Functionalities missing from the requirements

The steering controller could not be tested, because of magnetometer indoor disturbances.
The non-volatile storage could be solved with more time.

Functionalities missing from the requirements

The steering controller could not be tested, because of magnetometer indoor disturbances.
The non-volatile storage could be solved with more time.

What could we improve on this project?

- The belt vehicle, not enough control

Functionalities missing from the requirements

The steering controller could not be tested, because of magnetometer indoor disturbances.
The non-volatile storage could be solved with more time.

What could we improve on this project?

- The belt vehicle, not enough control
- The GoT system, good accuracy but not large range
- Route planning and initial mapping
- Blade control

Functionalities missing from the requirements

The steering controller could not be tested, because of magnetometer indoor disturbances.
The non-volatile storage could be solved with more time.

What could we improve on this project?

- The belt vehicle, not enough control
- The GoT system, good accuracy but not large range
- Route planning and initial mapping
- Blade control
- Other functionalities (humidity, obstacle, or length detection)