STATUS REPORT

Realization of an autonomous sail-boat

Written by:

Philippe Chuzel

ENSTA Bretagne

Option Systèmes Perception Information Décision

Contents

Abstract

Introduction

Part I

Regulation

Chapter 1

Sailboat regulation

The original algorithms which were used in order to regulate a sail-boat is given here [1].

For all those algorithms, we decide to chose one specific situation and we express the regulator in order to generate a potential adapted to the situation. For example, the following algorithm is used for the line regulation of the sail-boat 1.

Algorithm 1 Sail-boat regulator , line

Inputs

m position of the sailboat
θ heading of the sail-boat
ψ wind direction
a, b point of the line
q hysteresis

Output

δ_r ruder angle
δ_s sail angle
q hysteresis

Initialization

$q \leftarrow 1$

Algorithm

$e = det(\frac{b-a}{||b-a||}, m-a)$
if $|e| > \frac{r}{2}$ **then**
    q = sign(e)
end if
$\varphi = atan2(b-a)$
$\theta^* = \varphi - \frac{2\gamma_\infty}{\pi}.atan(\frac{e}{r})$
if cos(ψ-θ^*)+cos(ζ) < 0 or ($|e| < r$ and cos(ψ-φ) + cos(ζ)) **then**
    θ_p = π + ψ - q . ζ
else
    θ_p = θ^*
end if
δ_r = $\frac{\delta_s^{max}}{\pi}.atan(tan(\frac{\theta-\theta_p}{2}))$
δ_s = $\delta_s^{max}.\frac{cos(\psi-\theta_p)+1}{2}$

In the end, we generate a field potential as below which allow the sail-boat to follow the line.

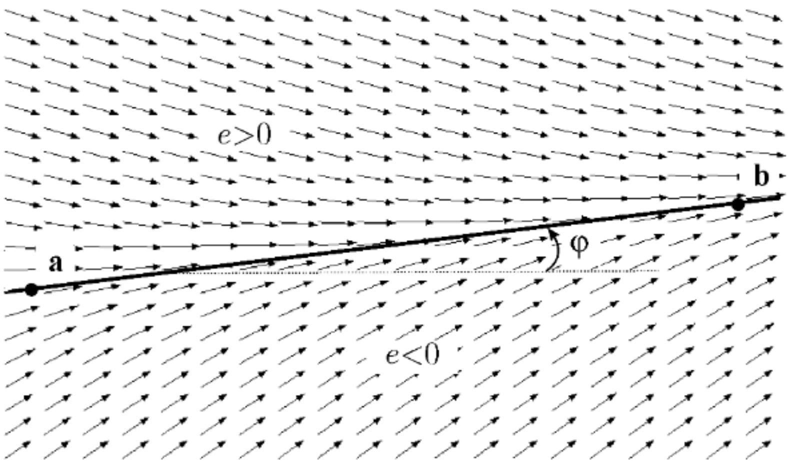

Figure 1.1: Field potential generated by this regulator.

Chapter 2

Generation of field potential

Nevertheless, we wanted to realize a new version of this algorithm which would allow us to get the value of the ruder angle and the sail angle by only providing the heading wanted, the real heading and the wind direction.

Chapter 3

update of the algorithm

The following algorithm is the result of our work 2.

Algorithm 2 Sail-boat regulator V1

Inputs

1: θ^* heading wanted
2: θ heading of the sail-boat
3: ψ wind direction
4: q hysteresis

Output

1: δ_r ruder angle
2: δ_s sail angle
3: q hysteresis

Initialization

1: $q \leftarrow 1$

Algorithm

1: **if** cos(ψ-θ^*)+cos(ζ) < 0 **then**
2: q = sign(sin($\psi - \theta^*$))
3: θ_p = π + ψ - q . ζ
4: **else**
5: θ_p = θ^*
6: **end if**
7: $\delta_r = \frac{\delta_s^{max}}{\pi}.atan(tan(\frac{\theta-\theta_p}{2}))$
8: $\delta_s = \delta_s^{max}.\frac{cos(\psi-\theta_p)+1}{2}$

Part II

A voir

Chapter 4

Presentation of the sail-boat

put photos here!!!

Chapter 5

Choice of the component

5.1 Rasberry Pi

Figure 5.1: Rasberry Pi nuke.

5.2 IMU Razor

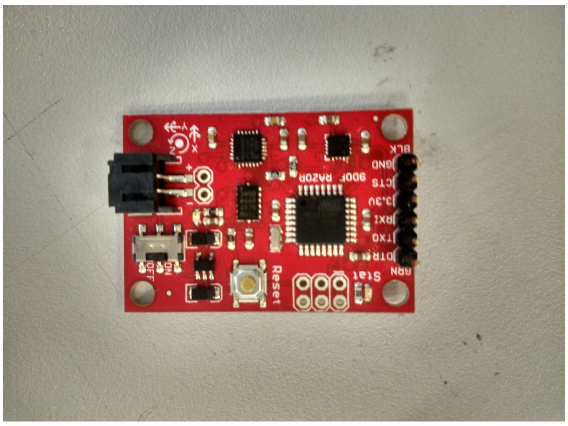

Figure 5.2: Razor IMU.

5.3 Magnetic stator, wind captor

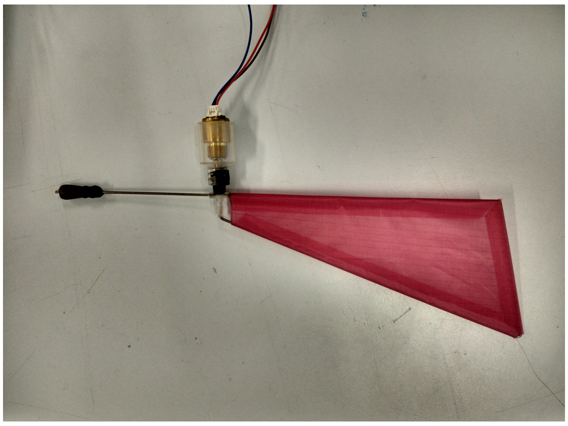

Figure 5.3: WindCaptor.

5.4 Servo-motor controller, pololu ship

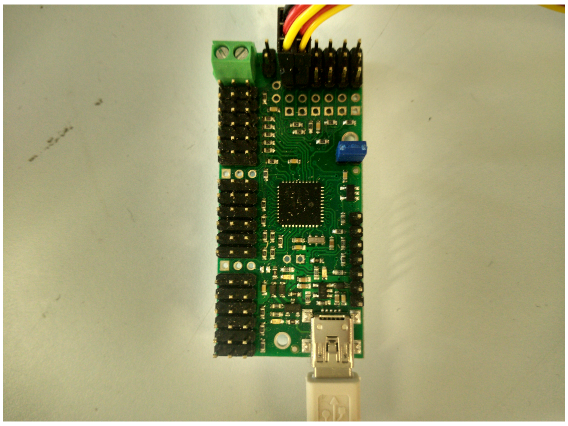

Figure 5.4: Pololu card.

5.5 GPS receiver

Figure 5.5: Example of GPS receiver.

Chapter 6

Global architecture

put photos here!!!

Part III

A voir

Part IV

Appendix

Bibliography

[1] Luc Jaulin. *La robotique mobile*. ISTE, 2015.

[2] Michel Kieffer, Luc Jaulin, Éric Walter, and Dominique Meizel. Robust autonomous robot localization using interval analysis. *Reliable computing*, 6(3):337–362, 2000.