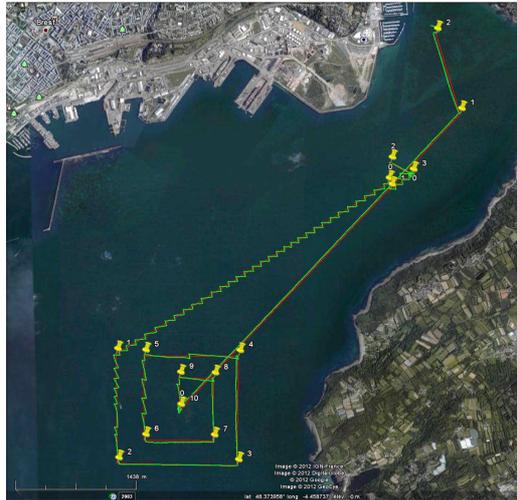


Commande d'un robot voilier (journée IHSEV)

1,2 juillet 2013.

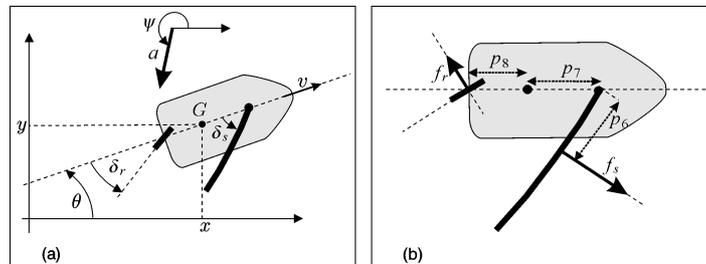
On considère un voilier qu'il nous faudra réguler suivant une ligne. L'idée est de reproduire des trajectoires du type de celle ci-dessous.



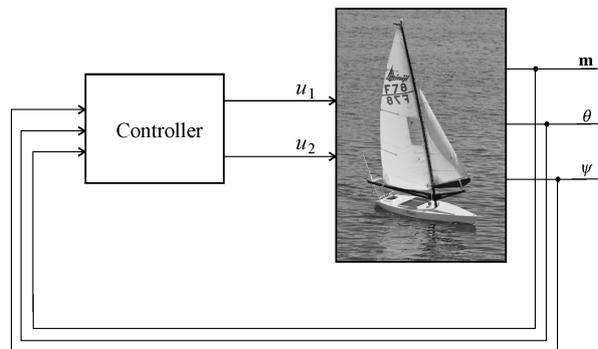
Rouge : ligne à suivre ; Vert : trajectoires faites par VAIMOS.

Entrées. Le robot possède deux entrées : $u_1 = \delta_r$ est l'angle du gouvernail, u_2 est la longueur de l'écoute.

Variables d'état. (x, y) sont les coordonnées du robot, θ est le cap, v est la vitesse tangentielle, ω est la vitesse de rotation.



Nous allons chercher à construire un régulateur comme illustré ci-dessous



1) Ajouter la ligne a suivre **ab** sur la scène. Il faudra pour cela rajouter les champs **ax, ay, bx, by** dans la classe **sailboat**. Pour l’affichage de la ligne, on pourra rajouter les lignes suivantes.

```
glBegin(GL_LINES);
glVertex3f(sailboat1.ax,sailboat1.ay,1);
glVertex3f(sailboat1.bx,sailboat1.by,1);
glEnd();
```

2) Rajouter à la classe **sailboat** une méthode **controller** qui permette de régler le gouvernail et la voile à travers les variables **deltag** et **deltavmax**.

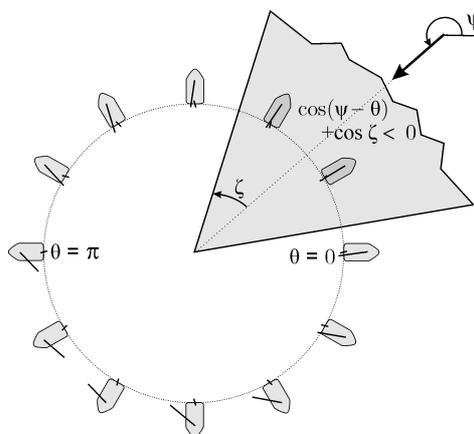
3) Supprimer la possibilité pour l’opérateur d’agir sur les commandes.

4) Implémenter le régulateur qui suit

Fonction in: $\mathbf{m}, \theta, \psi, \mathbf{a}, \mathbf{b}$; out: $\delta_r, \delta_s^{\max}$; inout: q	
1	$e = \frac{\det(\mathbf{b}-\mathbf{a}, \mathbf{m}-\mathbf{a})}{\ \mathbf{b}-\mathbf{a}\ }$
2	if $ e > r$ then $q = \text{sign}(e)$
3	$\varphi = \text{atan2}(\mathbf{b} - \mathbf{a})$
4	$\bar{\theta} = \varphi - \frac{1}{2} \cdot \text{atan}\left(\frac{e}{r}\right)$
5	if $\cos(\psi - \bar{\theta}) + \cos \zeta < 0$ then $\bar{\theta} = \pi + \psi - q \cdot \zeta$.
6	if $\cos(\theta - \bar{\theta}) \geq 0$ then $\delta_r = \delta_r^{\max} \cdot \sin(\theta - \bar{\theta})$ else $\delta_r = \delta_r^{\max} \cdot \text{sign}(\sin(\theta - \bar{\theta}))$
7	$\delta_s^{\max} = \frac{\pi}{2} \cdot \left(\frac{\cos(\psi - \bar{\theta}) + 1}{2} \right)$.

Le code correspondant est donné ci-dessous

```
void sailboat::Controller()
{ double r=10;
double zeta=M_PI/4;
double e=((bx-ax)*(y-ay)-(x-ax)*(by-ay))/hypot(ax-bx,ay-by);
if (e>r) q=1; if (e<-r) q=-1;
double phi=atan2(by-ay,bx-ax);
double thetabar=phi-0.5*atan(e/r);
if (cos(psi-thetabar)+cos(zeta)<0) thetabar=M_PI+psi-zeta*q;
double dtheta=theta-thetabar;
deltag=0.5*((cos(dtheta)>0)*sin(dtheta)+(cos(dtheta)<0)*((sin(dtheta)>0)-(sin(dtheta)<0)));
deltavmax=0.5*M_PI*(0.5*(cos(psi-thetabar)+1));}
```



Les directions infaisables forment la *no-go* zone.

