

Laboration 2 : Odometri och Bug-algoritmen

Rapport inkl. programkod lämnas in senast 8/4 kl 13.15.

Teorin för denna laboration finns beskriven i kompendiet (före Khepera-manualerna). Se även föreläsninganteckningarna samt laboration 1.

- Implementera odometri i en funktion `odometry` så att Kheperan kan hålla uppdatera sin "pose" $(x,y,theta)$. `odometry` läser av rotationsgivarna (wheel encoders) och returnerar den uppskattande positionen (x,y) och körriktningen $theta$.
Ex.: `[x,y theta] = odometry;`
Tips: funktionen `odometry` kan initialiseras med ett anrop med en start-pose som inparametrar. Använd `static` för att spara lokala variabler inne i funktionen `odometry`.
- Skriv en funktion `move2goal` som beräknar hastigheter $[v_l, v_r]$ som kan ta Kheperan i en rät (och rätt) linje från en punkt till en annan.
Ex.: `move2goal([fromX fromY theta], [toX toY])`
Testa rutinen med ett enkelt program.
- Implementera "the bug algorithm". Mål-positionen ska ges i inparametrar till funktionen:
Ex.: `bugalg([toX toY])`
Kheperan antas starta i $(0,0,0)$ och ska gå i riktning mot det angivna målet. Om den träffar på hinder går den över till en Breitenberg-vehicle som följer hindret tills den kan gå mot målet igen. När den nått målpunkten ska den stanna. Kodförslag:

```
function bugalg(goal)
comm_open;
position = odometry([0 0 0]);
<initialise>
while ~atgoal(position,goal)
    stimuli = read_prox;
    if obstruction(stimuli,position,goal)
        [Vl,Vr] = followobstacle(stimuli);
    else
        [Vl,Vr] = move2goal(position, goal);
    end
    set_speed([Vl Vr]);
    position = odometry;
end
set_speed([0 0]);
comm_close;
return % function bugalg
```

Roboten ska klara av situationer som dessa:

