1 Monte Carlo Lokalisering

Dette afsnit vil komme ind på, hvordan en autonom robot kan implementere en Monte Carlo Algoritme for således at kunne lokalisere sig selv på et kendt kort.

Der tages udgangspunkt i en autonom robot, der har fire hjul og mindst to elektriskemotorer med enkoder. Elektriskemotorer med enkoder afgiver et elektrisk signal n antal gange per total rotation af motorens akse. Dette kan eksempelvis være implementeret ved brug af en hall sensor, der måler på en magnet eller en optisk sensor, som måler på et optisk gitter der følger aksen. Således er robotten i stand til at udregne, hvor langt den har bevæget sig i en given retning. Denne længde udregnes i det følgende. Lad d være diameteren af robottens hjul,lad n være antallet af impulser per rotation, og lad til sidst n_t være et antal impulser målt efter en given tid. Længden L, som robotten har bevæget sig ved antal målte impulser n_t, kan således udregnes på følgende måde:

$$L = d \times \pi \times \frac{n_t}{n} \tag{1.1}$$

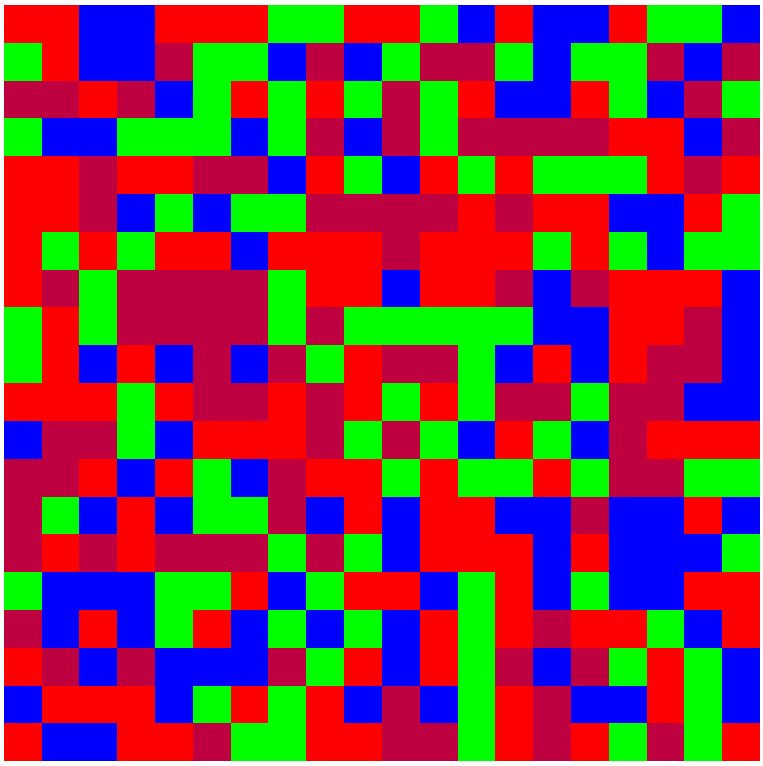
Et eksempel på overstående formel, kunne være at robotten havde et sæt hjul med diameteren $5cm$ altså $d = 5cm$ samt en hallsensor der giver 20 impulser per totale rotationer af aksen. Når robotten har bevæget sig 4512 impulser kan vi udregne den længde robotten har bevæget sig:

$$L = 5cm \times \pi \times \frac{4512}{20} = 3543{,}71cm = 35{,}4371m \tag{1.2}$$

Vi kan også udregne vinklen θ for robotten, da dette er en simpel implementering af Dead-reckoning vil vi kun udføre rotationer, når robotten står stille.

Desuden har robotten en sensor der er placeret under den, denne sensoren måler farven af hvad robotten bevæger sig på. Sensoren måler et resultat på formen (r, g, b), hvor $(0, 0, 0)$ vil være sort og $(255, 0, 0)$ ville være en rød farve.

Med et udgangspunk i overstående robot der implementer et Dead-reckoning algoritme, kan vi lokalisere robotten på et kendt kort ved brug af Monte Carlo Lokalisering. Det kendte kort i dette konkrete eksempel er et kort der er kvadratisk af 20×20 meter og er bestående af fliser med forskelig farver, disse farver er henholdsvis rød, grøn, blå og lilla, som illustreret på nedenstående figur.



Altså så vi har et kendt kort hvor der er 400 fliser af forskelige farvers hvilke også er punkter, dette repræsenteret vores udfaldsrum U, hvor dets elementer er punkter på formen $u_i = (x, y)$ og $rgb(u_i)$ er farven i RGB, altså hvis vi har en rød flise på punktet $(1, 2)$ vil $rgb((1, 2)) = (255, 0, 0)$. Dette udfaldsrum er en del af et symmetrisk sandsynlighedsfelt, hvilket betyder at robotten har en lige stor sandsynlighed for at befinde sig på et givent punk. Ud fra dette sandsynlighedsfelt kan vi definere en stokastisk variabel X som kan tage hvilket som helst punkt i udfaldsrummet U, hvor x_i er et konkrete punkt. Følgende nedenstående trin er derfor involverede i at lokalisere robotten på kortet.

1. n tilfældige konkrete stokastisk variabel x_i bestemmes.
2. Robotten laver en måling m på formen (r, g, b) på dens nuværende position
3. Målingen m sammenlignes nu med x_i på følgende måde $|m - x_i|$
4. Et nyt sandsynlighedsfelt λ hvor $P(x_i) = \frac{|m-x_i|}{s}$ hvor s er summen af alle fejl fra forgående trin og U er de forgående konkrete elementer x_i. Dette vil betyde at der er støre sandsynlighed for de punker med en stor fejl relativt til dem med en lav fejl
5. Der bestemmes nu λ_n tilfældige punkter fra λ, hvor $\lambda_n < n$, og fjernes fra λ_U
6. Robotten bevæger sig nu en flise i hvilken som helst retning, hvor robotten stadig vil være inden for fliserne. Alle resterende punkter i λ flyttes samme retning og længde
7. Der bestemmes nu λ_n, altså samme antal som blev fjernet, punktern fra et sandsynlighedsfelt hvor $P(x_i) = 1 - \frac{|m-x_i|}{s}$, altså så punker med en lav fejl har en højere sandsynlighed. En af de omliggende fliser fra dette punkt som robotten kunne have bevæget sig over på vælges og tilføjes til udfaldsrummet λ_u. Der efter gentages de forgående tre trin det antal gange robotten bevæger sig.

Hvis man fulgte overstående trin, vil man finde at man starter ud med en masse punker der ikke er koncentration hvor robotten rent faktisk befinder sig, men som processen gentages bliver der dannet en koncentration af punkter hvor robotten befinder sig, nedenstående figur illustreret dette.

figure

I nedenstående sektion, vil jeg komme ind på en teoretisk implementation af dette i Python.

1.1 implementering