Date
Page No.

MCL Algorithm.

Pseudocode.

Previous belief. Sensor measurement

Algorithm MCL $(\mathcal{X}_{t-1}, u_t, Z_t)$

Actuation command.

Initialize from sensor i/p. ← $\bar{\mathcal{X}}_t = \mathcal{X}_t = \phi$ (Later we refine this estimate)

Generate 'M' particles ← for $m = 1$ to M:

hypothetical state ← $x_t^{[m]}$ = motion-update $(u_t, x_{t-1}^{[m]})$

weight ← $w_t^{[m]}$ = sensor-update $(Z_t, x_t^{[m]})$

$$\bar{X}_t = \underbrace{\bar{X}_t}_{\text{prev state (belief)}} + \underbrace{\langle x_t^{[m]}, w_t^{[m]} \rangle}_{\text{motion \& sensor measurements}}$$

end for.

Resampling: particles with high probability survive & are redrawn in next iteration, while other particles die.

for $m = 1$ to M:

draw $x_t^{[m]}$ from $\bar{X}_t$ with probability $\propto w_t^{[m]}$

$X_t = X_t + x_t^{[m]}$

end for

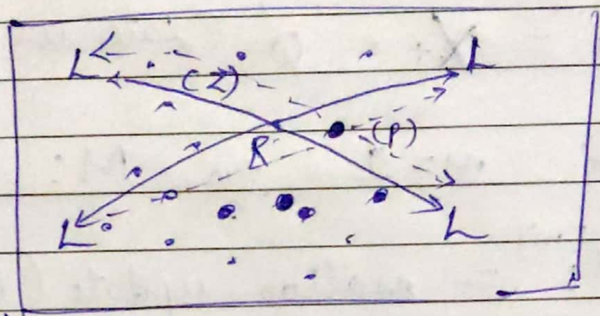
return X_t → O/p of new belief.

Date
Page No.

① Generate the particles (eg:- $N = 1000$)

② Randomly & uniformly spread the particles throughout the map.

③ Importance weight = Z−P



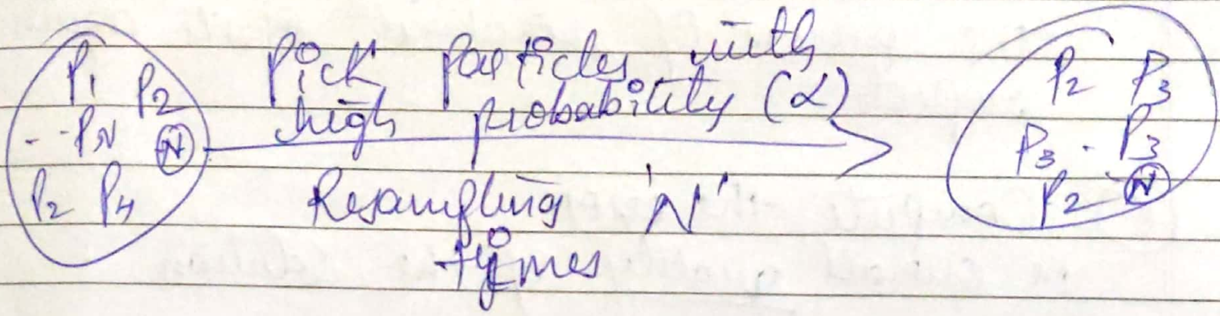
(sensor)
Z = actual measurement (Robot to landmarks)
P = Predicted " (Particle to landmarks)

→ Higher the weight, more important the particle is & more likely to represent the position of robot.

④ Resampling

	particles	Poses	Weights	Normalized weights.
N (1000)	P_1	(x_1, y_1, θ_1)	W_1	$\alpha_1 = W_1/W$
	P_2	(x_2, y_2, θ_2)	W_2	$\alpha_2 = W_2/W$
	⋮	⋮	⋮	⋮
	P_N.	(x_N, y_N, θ_N)	W_N	$\alpha_N = W_N/W$
			$W = \sum_i W_i$	$\sum_i \alpha_i = 1$

→ Randomly draw new particles from old once with replacement (i.e particles with high probability are drawn multiple number of times, so that @ any given point there would be $N=1000$ particles) in proportion to importance weight.



Eg: Suppose we have $N=6$ particles
P_1, P_2, P_3, P_4, P_5, P_6
and P_3 & P_6 have high weights than other
we will run resampling N (6) times.

I iteration = P_6 was chosen, again P_6 chosen, next P_3, P_3, P_5, P_1

So our particle list at I iteration look like P_6, P_6, P_3, P_3, P_5, P_1

after N iteration = the list may look like
P_6, P_6, P_6, P_6, P_6, P_6.

and we see all particles approximately at the car's location & can say that our car is localized.

Date
Page No.

✓ This was just for one frame of instant, In the next instant the car moves so do the particles,

These particles are once from the previous step of resampling.

The process of localizatn starts again & repeats.

(5) Compute the error
ie overall quality of the solution

computing the avg distance betn the particle & the Robot.

ie difference between the x & y posn of robot with x & y posn of ~~robot~~ particle

~~also normalizing the values~~.

Next, Compute the euclidian distance of the difference

ie $dx = part.x - rob.x$
$dy = part.y - rob.y$.
$error = \sqrt{dx^2 + dy^2}$
$sum += error$

return sum / N.