

Universidade de Coimbra

Projeto - I
Fusão de Informação em Análise de Dados
Professor Alberto Cardoso
Professor Jorge Henriques

Gonçalo Paiva Gouveia 2018277419

11 de abril de 2022

1 Modelos

Os filtros de Kalman e de Partículas são algoritmos que atualizam recursivamente uma estimativa do estado e encontram as inovações que conduzem um processo estocástico a partir de uma sequência de observações. O filtro de Kalman atinge esse objetivo por projeções lineares, enquanto o filtro de partículas o faz pelo método de Monte Carlo sequencial.

1.1 Dead Reckoning (DR)

Na navegação, DR é o processo de calcular a posição atual de algum objeto em movimento usando uma posição previamente determinada ou fixa e, em seguida, incorporando estimativas de velocidade, sentido, direção e curso ao longo do tempo decorrido.

Dead Reckoning está sujeito a erros cumulativos. No entanto, os sistemas de navegação inercial, que fornecem informações direcionais muito precisas, usam cálculos mortos e são amplamente aplicados.

1.2 Extended Kalman Filter (EKF)

É baseado em uma linearização usando uma aproximação em série de Taylor dos modelos dinâmicos e de medição não lineares. Usando essa linearização, são encontradas etapas de atualização de previsão e medição semelhantes ao filtro de Kalman.

este é essencialmente o mesmo algoritmo que o KF original onde a função não linear toma o lugar da predição (tanto na etapa de predição quanto na predição da saída) e nas atualizações de matrix de covariância. Além disso, se um dos modelos for linear, a etapa correspondente (previsão ou atualização de medição) pode ser substituída por uma etapa de atualização exata do filtro de Kalman

Um dos principais problemas do EKF é que a linearização é local, o que muitas vezes leva a problemas .

1.3 Unscented Kalman Filter (UKF)

O Unscented Kalman Filter é também um algoritmo para modelos nãolineares, contudo, são geralmente mais precisos, pois são gerados sigma-points (SP) e respetivos pesos, que depois de sofreram as transformações que o modelo real sofreu, acabam a ser mais próximos da realidade, seguindo também aproximadamente uma distribuição gaussiana. Depois, apenas trata de prever os valores tal como no Kalman Filter.

Aqui a principal diferença do EKF é que no EKF pegamos apenas um ponto, ou seja, média e aproximada, mas no UKF pegamos um monte de pontos chamados pontos sigma e aproximamos com o fato de que quanto mais pontos, mais precisa nossa aproximação será!

1.4 Particle Filter (PF)

Este algoritmo gera vários pontos e aplica o mesmo movimento até chegar à posição real do robot. Segue-se uma descrição mais detalhada do algoritmo:

- 1. Começamos por criar alguns landmarks para posteriores comparações
- De seguida, são geradas várias partículas em posições aleatórias no espaço de procura. Todas as partículas têm a mesma probabilidade de estar no sítio certo
- 3. É então comparada a posição real (obtida de dados GPS por exemplo) com todas as partículas geradas (através da distâncias aos nossos pontos de referência) e as partículas que tiverem em posições semelhantes, terão assim um maior peso.
- 4. Nas posições onde concluímos que é mais provável corresponder à realidade, geramos novas partículas nessas regiões.
- 5. O robot move-se e aplicamos o mesmo movimento às partículas.

Repetindo os passos de 3 a 5, ao longo das iterações a tendência das partículas é a de se aproximarem a realidade.

2 Resultados dos filtros aplicados

- A linha azul é a trajetória verdadeira.
- A linha preta é a trajetória por posicionamento relativo (dead-reckoning).
- OS pontos verdes correspondem à observação do posicionamento do robot (por ex. GNSS).
- A linha vermelha é a trajetória estimada com o EKF.
- A linha ciano corresponde a trajectória com o Filtro de partículas.
- A elipse vermelha a tracejado representa a elipse de covariância estimada com o EKF.

2.1 EKF e UKF

Do código desenvolvido para este projeto (em anexo) foi possível obter o seguinte gráfico:

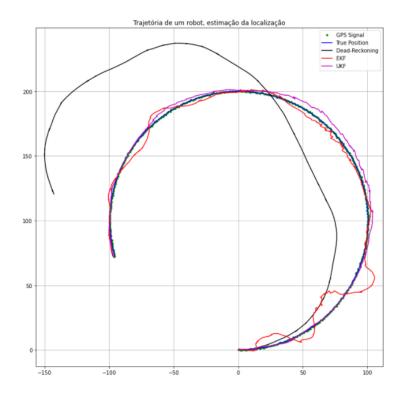


Figura 1: Estimação da localização de um robot.

Fazendo varias simulações é possível obter a estimação dos seguintes parâmetros (Parâmetros estimados com 100 iterações).

	MSE_mean	MSE_std	Absolute Error	RMSE_mean	RMSE_std	R^2_mean	R^2_std
Dead Reckonin	d- g 2130.072971	2282.681770	0.071645	39.435064	19.563931	0.553396	0.468989
EK	F 11.025091	7.233790	0.343879	3.162306	0.980505	0.997633	0.001551
UK	F 9.137986	7.932434	0.131928	2.811110	1.057049	0.998071	0.001659

Figura 2: Estimadores obtidos para cada modelo (DR,UKF, EKF).

2.2 PF

Do código desenvolvido para este projeto (em anexo) foi possível obter o seguinte gráfico:

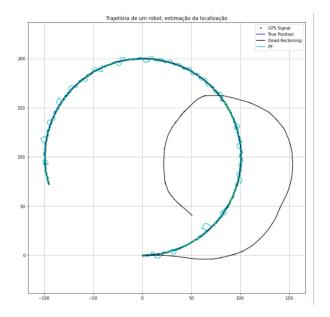


Figura 3: Exemplo de imagem obtida para o método PF.

Fazendo a análise de performance em função do número de sementes escolhida (número de partículas constante igual a 1000), e o número de partículas escolhido em função do número de seeds constante (igual a 10).

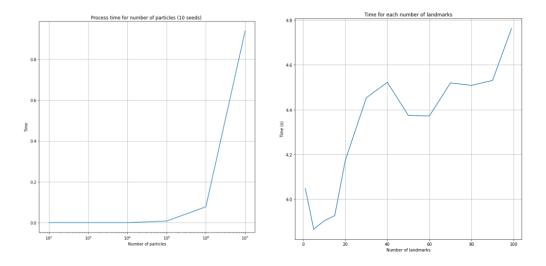


Figura 4:

3 Discussão

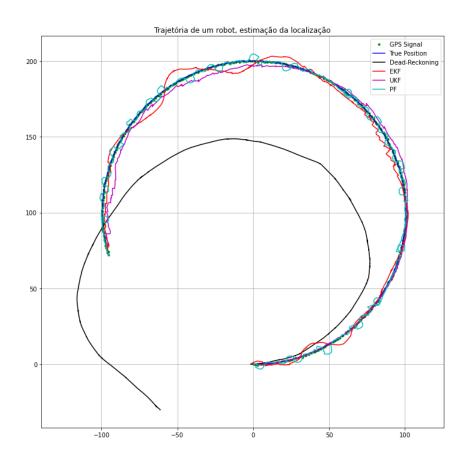


Figura 5: Trajectórias dos diferentes modelo (DR, UKF, EKF, PF).

		MSE_mean	MSE_std	Absolute Error	RMSE_mean	RMSE_std	R^2_mean
	Dead-Reckoning	2043.588042	1778.033430	0.129945	39.023097	19.954674	0.558627
	EKF	12.010627	5.127573	0.573080	3.382016	0.696392	0.997416
	UKF	7.280795	3.912474	0.462631	2.593390	0.661530	0.998446
	Particle Filter	4.667107	1778.033430	0.913381	2.154032	0.092679	0.999007

Figura 6: Estimadores obtidos para cada modelo (UKF, EKF, PF).

- O algoritmo com os valores mais baixos de MSE e RMSE é o Particle Filter, contudo, conseguimos perceber que é instável (pelo MSEstd).
- EKF, UKF e Particle Filter têm resultados bastante próximos da reali-

dade, Podemos tirar esta conclusão quando olhamos para os valores de teste de qualidade do ajuste (R^2) .

- Sem sombra de dúvidas, quando comparamos os vários modelos o pior deles todos é o Dead-Reckoning (analisando as várias métricas). Dead Reckoning esta sujeito a erros cumulativos.
- Os algoritmos mais estáveis são o EKF e UKF pois o ${\it MSE}_{std}$ tem os valores mais baixos.
- Quando se avaliamos a performance do Filtro de partículas, sendo um método baseado em Monte Carlo, quanto maior for o número de partículas,e o número de Seed, maior é o tempo de execução do método. Mas também maior é a convergência para resultado final, nos casos mais genéricos.

4 Conclusão

Posso concluir que as tarefas delineadas para este projecto foram concluídas com êxito. Aprendendo como implementar vários modelos de fusão de dados sensoriais.