



UNIVERSIDADE
ESTADUAL DE LONDRINA

ENZO GUIDO AMERICANO DA COSTA

MONITORAMENTO E PREDIÇÃO DE SISTEMAS
AUTÔNOMOS

LONDRINA

2023

ENZO GUIDO AMERICANO DA COSTA

**MONITORAMENTO E PREDIÇÃO DE SISTEMAS
AUTÔNOMOS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

Orientador: Prof Adilson Luiz Bonifacio

Coorientador: -

LONDRINA

2023

ENZO GUIDO AMERICANO DA COSTA

**MONITORAMENTO E PREDIÇÃO DE SISTEMAS
AUTÔNOMOS**

Trabalho de Conclusão de Curso apresentado ao curso de Bacharelado em Ciência da Computação da Universidade Estadual de Londrina para obtenção do título de Bacharel em Ciência da Computação.

BANCA EXAMINADORA

Adilson Luiz Bonifácio

Orientador: Prof Adilson Luiz Bonifacio
Universidade Estadual de Londrina

-

-

-

-

-

-

Londrina, - de 2023.

*Este trabalho é dedicado às crianças adultas
que, quando pequenas, sonharam em se
tornar cientistas.*

AGRADECIMENTOS

- . **Monitoramento e predição de sistemas autônomos**. 2023. 54f. Trabalho de Conclusão de Curso (Bacharelado em Ciência da Computação) – Universidade Estadual de Londrina, Londrina, 2023.

RESUMO

O objetivo dessa pesquisa é unir informações que geralmente se encontram separadas e segmentadas em diversos artigos e outras pesquisas relacionadas com o tema de localização e mapeamento simultâneos (SLAM), com o intuito de entender o que esse desafio proporciona como solução e avanço sobre diversos quesitos, desde comerciais e empresariais até utilidades domésticas. Além disso, todos os tópicos desse tema serão destrinchados e explicados de forma isolada para, assim, demonstrar uma simulação de aplicação real utilizando todos os meios e métodos descritos.

Palavras-chave: SLAM. Filtro de Kalman. Sistemas autônomos

- . **Monitoring and prediction of autonomous systems**. 2023. 54p. Final Project (Bachelor of Science in Computer Science) – State University of Londrina, Londrina, 2023.

ABSTRACT

The goal of this research is to unite information that is usually splited and segmented in several articles and other researches related to the topic of simultaneous localization and mapping (SLAM), in order to understand what this challenge provides as a solution and advance on several issues, from commercial and business to housewares. In addition, all topics of this theme will be unraveled and explained separately, thus demonstrating a simulation of a real application using all the means and methods described.

Keywords: SLAM. Kalman filter. Autonomôus systems

LISTA DE ILUSTRAÇÕES

Figura 1 – Exemplo de um sensor LIDAR 2D [23].	17
Figura 2 – Exemplo de funcionamento ICP 3D [26]	18
Figura 3 – Representação dos intervalos de confiança sobre uma distribuição normal gaussiana que correlaciona duas variáveis (x e y) [27].	21
Figura 4 – Etapas do filtro de Kalman [20].	27
Figura 5 – Resultados do exemplo prático com filtro de Kalman multidimensional	30
Figura 6 – Resultados do exemplo prático com filtro de Kalman bidimensional	47
Figura 7 – Leitura planejada de um LIDAR 2D em um ambiente 3D.	48

LISTA DE TABELAS

LISTA DE ABREVIATURAS E SIGLAS

TCC Trabalho de Conclusão de Curso

SUMÁRIO

1	INTRODUÇÃO	11
1.1	Histórico	12
1.2	Contextualização do problema	13
1.3	Organização do Trabalho	14
2	FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA	15
2.1	Arquitetura do SLAM	16
2.1.1	Sensores e o Sistema ICP	16
2.1.2	Filtro de Kalman	19
2.2	O Sistema SLAM Unidimensional	19
2.2.1	Conceitos e Notações	20
2.2.2	Modelo de Predição Unidimensional	24
2.2.3	Aplicação do filtro de Kalman unidimensional	26
2.3	O Sistema SLAM Multivariado	30
2.3.1	Modelo de predição Bidimensional	31
2.3.2	Aplicação prática do filtro de Kalman multivariado	39
2.3.3	Projetando o Ambiente Tridimensional para Bidimensional	47
2.4	Algoritmos e estrutura de código	48
	REFERÊNCIAS	52

1 INTRODUÇÃO

Nos últimos anos, temos testemunhado um avanço significativo na área de sistemas embarcados, robótica e inteligência artificial, impulsionado pela demanda por sistemas autônomos capazes de realizar tarefas complexas, tais como [1, 2, 3]. Um dos principais desafios enfrentados por esses sistemas é a necessidade e habilidade de compreender e interagir com o ambiente ao seu redor. O bom funcionamento desses sistemas depende de alguns aspectos importantes relacionados a arquitetura entre seus diferentes componentes de forma organizada e coordenada. Essa arquitetura é composta de partes mecânicas, tais como rodas e motores para locomoção, bem como dispositivos digitais tais como os sensores que permitem a análise do ambiente externo, computadores e controladores para realizar as atividades propostas, além da comunicação entre as partes e, por fim, uma estrutura externa capaz de armazenar todas as partes em um único modelo completo.

Nesse contexto, os algoritmos de predição de movimento, ou SLAM¹, têm se destacado como uma solução promissora a fim de orquestrar todos esses componentes em prol de um objetivo específico. O SLAM [4, 5] é uma técnica que combina a capacidade de mapeamento de um ambiente desconhecido e a localização simultânea do agente autônomo dentro desse ambiente. Em outras palavras, o algoritmo SLAM permite que um robô, veículo autônomo ou qualquer outro sistema móvel, obtenha informações sobre o ambiente e, ao mesmo tempo, determine sua própria posição e orientação dentro desse ambiente.

A necessidade de algoritmos SLAM surge em diversos contextos, tais como robótica móvel, realidade aumentada, veículos autônomos, drones, aplicações industriais e até mesmo uso pessoal como aspiradores de pó automáticos. Esses algoritmos são fundamentais para que os sistemas autônomos possam navegar em ambientes desconhecidos ou em constante mudança, construindo mapas detalhados enquanto se localizam com precisão.

As primeiras ideias para lidar com sistemas autônomos surgiram com os algoritmos de filtro de Wiener [6] e Kalman [7], baseados na estatística probabilística, para estimar, de forma precisa, a posição e orientação dos objetos de estudo, com uma fonte de dados limitada e imprecisa que provém do ambiente externo. O filtro de Kalman, então, começou a ser aplicado como técnica de estimativa de estado no problema do SLAM [8, 9]. Isso levou ao desenvolvimento do SLAM baseado em filtro de Kalman como conhecemos atualmente.

¹ do inglês, *Simultaneous Localization and Mapping*

1.1 Histórico

O SLAM surgiu durante na Conferência de Robótica e Automação do IEEE em 1986. Naquela época, os métodos baseados em probabilidade estavam sendo gradualmente introduzidos nas áreas de robótica e inteligência artificial, como a proposta de Rudolf E. Kalman [7]. Kalman propôs um método estatístico, que ficou conhecido como filtro de Kalman, para estimar o estado de um sistema dinâmico em tempo real.

Diversos pesquisadores então começaram a estudar sobre técnicas estimativas para resolver questões relacionadas à mapeamento e localização, tais como Cheeseman e Smith [10], que propuseram um método para relacionar o erro estimado entre localizações relativas de objetos, Durrant-Whyte [11], que reuniu diversos modelos essenciais com implementações e demonstrações de métodos e algoritmos para solucionar o problema SLAM, Chatila [12], que discutiu problemas e possíveis soluções para mapeamento utilizando robôs autônomos, como perda de referência, ambiente volátil e condições irregulares, entre outros que também desempenharam papéis valiosos, contribuindo de maneira significativa para o desenvolvimento inicial de um problema que, mais tarde, foi considerado um dos maiores avanços na área de robótica móvel e autônoma.

O problema do SLAM ganhou destaque à medida que os avanços na robótica começaram a permitir que máquinas interagissem de maneira mais autônoma com o mundo. No início, a robótica se baseava em ambientes controlados onde os robôs seguiam rotas predefinidas, como robôs AGV que seguem faixas demarcadas e apenas podem percorrer sobre elas [13] ou eram comandados remotamente [14]. Entretanto, à medida que as aplicações robóticas se expandiram para ambientes não estruturados, mutáveis e desconhecidos, tornou-se essencial que os robôs fossem capazes de se localizar por conta própria e navegar de forma independente.

Uma das soluções para a captura de dados externos do ambiente, tais como ângulo referencial, distância percorrida e até reconhecimento de estruturas em planos 2D e 3D, foi o uso de sensores. Os dados fornecidos por sensores são reconhecidos em algoritmos e procedimentos que os transformam em equações matemáticas que resumem o ambiente, os movimentos do robô autônomo, sua posição e orientação. A abordagem probabilística é frequentemente usada para tratar a incerteza associada às medições dos sensores, às estimativas e predições.

Os primeiros trabalhos sobre SLAM exploraram métodos para resolver esse problema através de modelos estatísticos que pudessem representar incerteza nas estimativas de localização e mapeamento. Os algoritmos, como o filtro de Kalman estendido (EKF) [15], foram aplicados inicialmente para combinar dados dos sensores e estimativas de movimento em robôs a fim de atualizar o mapa e a localização.

Com o avanço das pesquisas abordagens mais sofisticadas foram surgindo, como

os Filtros de Partículas (Particle Filters) [16] e o uso de técnicas de otimização como do SLAM baseado em grafo [17]. Além disso, a crescente disponibilidade de sensores avançados, como câmeras 3D [18] e LIDAR [19], possibilitou melhorias significativas na qualidade e detalhes dos mapas gerados pelo SLAM. O desenvolvimento deste trabalho ficará mais focado nos algoritmos de predição, bem como a utilização do sensor LIDAR para mapeamentos 2D, juntamente com o filtro de Kalman [20].

1.2 Contextualização do problema

Sistemas autônomos são capazes de realizar tarefas e tomar decisões de forma independente, sem a necessidade de intervenção humana direta. Um desafio crucial para esses sistemas está na obtenção de informações precisas sobre o ambiente para estimar sua posição e orientação em relação a esse ambiente. Neste cenário o filtro de Kalman desempenha um papel fundamental, particularmente em sistemas onde os dados de entrada são afetados por erros de medição e ruído, como é o caso de sensores utilizados em sistemas autônomos.

O filtro de Kalman funciona combinando duas fontes de informações: as medições e as previsões do estado do sistema. As medições são fornecidas pelos sensores, câmeras, radares, entre outros dispositivos que capturam informações do ambiente. A partir desses dados, previsões do estado do sistema são obtidas por meio de um modelo matemático que descreve o comportamento dinâmico do sistema. O filtro de Kalman combina essas informações de maneira ponderada, levando em consideração as incertezas associadas a cada uma delas.

Em sistemas autônomos, o filtro de Kalman é frequentemente usado para estimar a posição, orientação e velocidade do veículo ou robô em relação a um mapa do ambiente. Os sensores, como GPS, giroscópios, acelerômetros e câmeras, fornecem informações sobre a movimentação e percepção do sistema. Essas informações são utilizadas pelo filtro para atualizar continuamente a estimativa do estado do sistema, permitindo que o sistema autônomo entenda sua posição em relação ao ambiente e tome decisões apropriadas. Neste sentido o SLAM desempenha um papel crucial na percepção do ambiente, permitindo que os veículos tomem decisões precisas e seguras em tempo real.

Portanto, os algoritmos SLAM representam uma solução revolucionária para a navegação e mapeamento autônomos, proporcionando a capacidade de explorar, compreender e interagir com o ambiente de maneira eficiente e precisa. Com o contínuo progresso nessa área, podemos esperar um futuro em que sistemas autônomos desempenhem um papel cada vez mais significativo em nossa sociedade, trazendo benefícios nas áreas de transporte, logística, segurança e muito mais.

1.3 Organização do Trabalho

O objetivo deste trabalho é então juntar diversos conceitos e conhecimentos, muitas vezes desconexos, relativos ao tema de monitoramento e predição de sistemas autônomos, afim de propor uma possível solução para o problema de localização autônoma na área da robótica. Essa solução que envolve apenas os mecanismos internos na construção de um robô que possa utilizar dos resultados aqui obtidos, diferente de outros modelos que necessitam de um sistema externo para se localizar em um ambiente, como é o caso dos robôs AGV já citados na introdução [13, 14].

Dessa forma, para a construção e desenvolvimento desta solução, o trabalho em questão foi organizado em três capítulos, sendo o primeiro a Introdução, este capítulo que tem o objetivo de apresentar o modelo que foi estudado e escolhido como base para a solução final, contendo seu histórico e os motivos pelo qual foi utilizado, e a contextualização do problema, que mostra como surgiu a necessidade de um conceito/produto que possa suprir as demandas tanto domésticas quanto industriais, seguido desta seção que apresenta essa organização.

O segundo capítulo é relativo a Fundamentação Teórico-Methodológica, é nele que a maior parte do conteúdo sobre o tema é apresentado e percorrido, estando presente a arquitetura do modelo, onde é mostrado como o modelo em si pode ser estruturado e os aparelhos necessários para sua aplicação, além da apresentação do conceito de algoritmo base que será utilizado para a resolução do problema; a apresentação do modelo esquematizado para um ambiente unidimensional, essa que serve para introduzir os conceitos e notações que serão utilizados no desenvolvimento da aplicação, os cálculos necessários e suas devidas derivações, e um exemplo demonstrativo de como e quando é possível realizar a localização de um objeto de estudo sob um ambiente descrito; a apresentação do modelo esquematizado para um ambiente multivariado, mais especificamente bidimensional, onde os conceitos e cálculos previamente apresentados são modificados para ser possível sua utilização em um modelo bidimensional, utilizando transformações matriciais para lidar com as múltiplas dimensões, outro exemplo demonstrativo da utilização deste modelo agora bidimensional, e como projetar essa solução bidimensional para ser utilizada em um ambiente real tridimensional; e por fim nesse capítulo é apresentado a seção de algoritmos e estrutura de código, onde nela estão os pseudo-códigos que servirão de base para a simulação desenvolvida no próximo capítulo.

Por fim o terceiro e último capítulo é relativo ao desenvolvimento de uma simulação, tanto do ambiente quanto dos sensores, projetada para ser totalmente funcional em um ambiente real tridimensional, ela utiliza do modelo bidimensional, e sua projeção para a resolução do problema, e consegue efetivamente realizar a localização de um objeto de estudo a partir dos conceitos e estudos apresentados.

2 FUNDAMENTAÇÃO TEÓRICO-METODOLÓGICA

Este capítulo tem o objetivo de estabelecer uma base sólida dos tópicos abordados ao longo deste trabalho, tais como conceitos, notações e formulações fundamentais sobre os modelos de previsão. Conhecimentos essenciais relacionados à estatística probabilística, especialmente àqueles que trata dos efeitos de ruídos em dados, também são abordados neste capítulo.

A contextualização inicial visa fornecer um panorama claro e abrangente do problema que será abordado. Serão delineados os cenários em que a análise probabilística se torna essencial, apontando também as situações em que os ambientes não apresentam estabilidade e, ao contrário, exibem mudanças constantes ao longo do tempo. Essa variabilidade e dinâmica apresentam desafios significativos na obtenção de informações precisas e na tomada de decisões assertivas. A compreensão do problema nesse contexto é fundamental para a posterior aplicação dos modelos probabilísticos.

Para viabilizar a modelagem estatística desses ambientes em constante transformação, será apresentado um conjunto de conceitos-chave, juntamente com as notações e equações associadas. A base teórica abordará aspectos unidimensionais e multivariados, permitindo que diferentes níveis de complexidade e aplicação sejam explorados. A clareza nessa etapa é crucial para garantir uma compreensão sólida das ferramentas estatísticas a serem empregadas.

Apresentados a contextualização inicial e os conceitos-chave é então introduzido o modelo de solução SLAM unidimensional, nele são apresentados os cálculos e derivações necessários para o entendimento e desenvolvimento completo do sistema. Em sequência é apresentado o modelo de solução SLAM multivariado, ele que utiliza das derivações e cálculos mencionados na seção anterior e vai além, apresentando as modificações necessárias para o funcionamento do modelo agora multivariado, com transformações matriciais, e uma transformação de um ambiente real tridimensional para um ambiente bidimensional e, dessa forma, ser possível utilizar da solução apresentada em situações reais.

Por fim, para ilustrar e consolidar o entendimento dos conceitos apresentados nessa seção, serão oferecidos exemplos elucidativos de como essas abordagens probabilísticas operam em cenários simulados, tanto em soluções unidimensionais como bidimensionais. Serão criadas situações fictícias em que a dinâmica de ambientes mutáveis é evidenciada. Através desses exemplos, os leitores poderão visualizar como os modelos probabilísticos são aplicados na prática, permitindo que a teoria se transforme em compreensão concreta.

2.1 Arquitetura do SLAM

A arquitetura do SLAM é composta pelo sistema de predição, propriamente dito, com o filtro de Kalman, e de outros componentes/sistemas auxiliares. Sensores, por exemplo, são componentes importantes que fazem parte do SLAM e servem para obtenção de dados em tempo real. Já o ICP é um sistema auxiliar que trata os dados obtidos e os transforma nos dados a serem manipulados pelo filtro de Kalman. Essa arquitetura é essencial para aplicações que envolvem desde rastreamento de automóveis até navegação autônoma com modelagem de mapa e localização em tempo real.

2.1.1 Sensores e o Sistema ICP

Os algoritmos SLAM utilizam uma grande variedade de sensores para coletar dados do ambiente, para que sejam processados através técnicas de visão e geometria computacional, processamento de imagens e filtros. O objetivo é extrair informações significativas do ambiente, identificar obstáculos, determinar a posição do agente autônomo e criar um mapa consistente e atualizado que possa se adaptar a variações no ambiente [21].

Os sensores LIDAR¹ [22], por exemplo, são amplamente usados em aplicações de robótica móvel, veículos autônomos, mapeamento topográfico, monitoramento ambiental, entre outros. Esse tipo de sensor fornece uma maneira precisa e rápida de obter informações detalhadas sobre o ambiente, permitindo que os sistemas autônomos tomem decisões confiáveis. Um sensor LIDAR é um dispositivo que usa emissão de lasers para medir distâncias e obter informações detalhadas sobre a geometria e características de um ambiente tridimensional. Por isso, o princípio de um sensor LIDAR é semelhante ao de um radar, que se utiliza das ondas de rádio, ao invés de feixes de luz.

O funcionamento de um sensor LIDAR, de maneira mais precisa, consiste na emissão de um pulso curto e intenso de luz dentro do ambiente, com uma duração e potência que podem variar dependendo do sensor. O pulso de luz viaja pelo ambiente até atingir um objeto, como uma parede, uma árvore ou qualquer outra superfície refletora, quando o pulso de luz atinge um objeto, uma parte da luz é refletida em direção ao sensor LIDAR, contendo informações sobre a distância aproximada do objeto. O sensor possui um detector sensível à luz que registra o pulso de retorno. Como o tempo entre a emissão do pulso e a detecção do pulso refletido é medido com alta precisão, o sensor LIDAR calcula a distância entre o sensor e o objeto com base nos tempos de ida e volta do pulso, e na constante da velocidade da luz. A emissão dos pulsos de luz ocorrem repetidamente e em diferentes direções, seja por meio de um scanner giratório ou de uma matriz de feixes de luz, permitindo a construção detalhada de um mapa bidimensional, ou até mesmo tridimensional, do ambiente.

¹ do inglês, *Light Detection and Ranging*

A Figura 1 demonstra o funcionamento de um sensor LIDAR 2D. Os dados coleta-

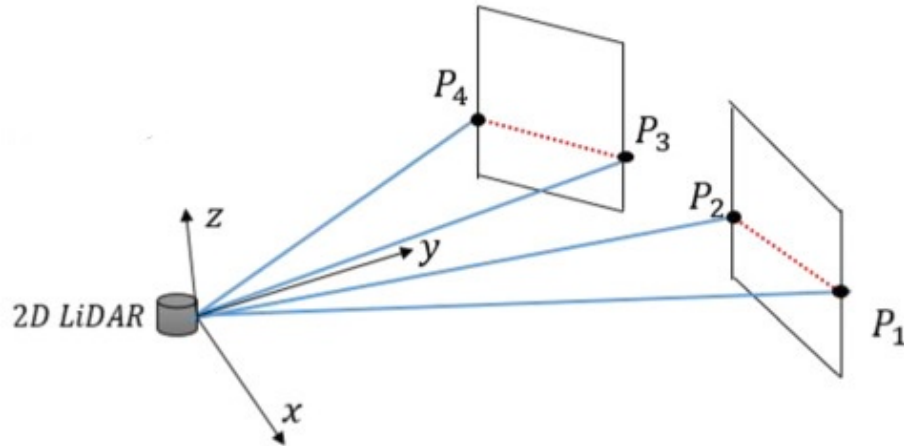


Figura 1 – Exemplo de um sensor LIDAR 2D [23].

dos pelo sensor LIDAR são retornados na forma de pontos no plano cartesiano (coordenadas x e y no plano 2D ou x , y e z no plano 3D) e processados por um algoritmo que remove ruídos, filtra pontos inválidos e obtém informações úteis, como a posição tridimensional dos objetos, sua forma, tamanho e intensidade de retorno [24].

Algumas aplicações mais atuais sobre SLAM têm usado métodos de processamento de imagens e inteligência artificial com câmeras RGB de profundidade para obtenção dos dados externos e permitindo o mapeamento 3D, como no caso dos algoritmos SLAM em carros TESLA [25]. No entanto, os sensores LIDAR ainda são largamente usados devido a sua maior eficiência em ambientes menores e menos voláteis que exijam apenas um mapeamento 2D, como nos casos de residências e galpões de logística. O menor custo em relação às câmeras RGB de profundidade, maior estabilidade e confiança nos dados (em um plano bidimensional) e menor custo computacional, não exigindo um alto poder de processamento e uma base de dados muito grande, são fatores que reforçam a aplicação dos sensores LIDAR em soluções SLAM.

Os sensores, em geral, retornam apenas pontos no plano cartesiano correspondentes à posição de um objeto. Daí a importância do ICP², um algoritmo que transforma os dados de entrada dos sensores para o filtro de Kalman, obtendo os parâmetros de translação, rotação, orientação, velocidade e aceleração do objeto de estudo. Essa transformação se dá pelo ajuste da translação e rotação sobre uma nuvem de pontos, tridimensionais ou bidimensionais, como as obtidas a partir de um sensor LIDAR, para melhor alinhar com uma outra nuvem de pontos de referência, ou seja, quanto uma nuvem precisa transladar e rotacionar para que esteja na mesma posição e orientação de outra nuvem base.

² do inglês, *Iterative Closest Point*

Ao comparar a nuvem de pontos capturada pelo LIDAR com uma nuvem de pontos de referência prévia (por iteração), o ICP então procura a transformação, rotação e translação, que melhor alinha os pontos de ambas as nuvens. Essa transformação é estimada iterativamente, refinando o resultado a cada iteração até convergir para uma estimativa mais precisa e ajustada de acordo com as necessidades do sistema. A estimativa dessa transformação é obtida pelo ICP através da otimização de uma função objetiva de minimização da diferença entre os pontos correspondentes das duas nuvens.

A Figura 2 mostra o funcionamento do ICP na prática. A nuvem de pontos na cor preta representa a leitura base, e o objetivo é alinhar a outra nuvem de pontos em relação a essa base. Os pontos na cor verde representa a primeira leitura do estado do objeto, que se encontra deslocada em relação a leitura base. Após uma iteração a nuvem de cor verde se torna vermelha, indicando que houve um deslocamento para um estado mais próximo do desejado. A cada iteração as nuvens de pontos tendem a ter uma distância cada vez menor, onde o usuário pode definir o grau de exatidão necessário com base no número de iterações que devem ser realizadas.

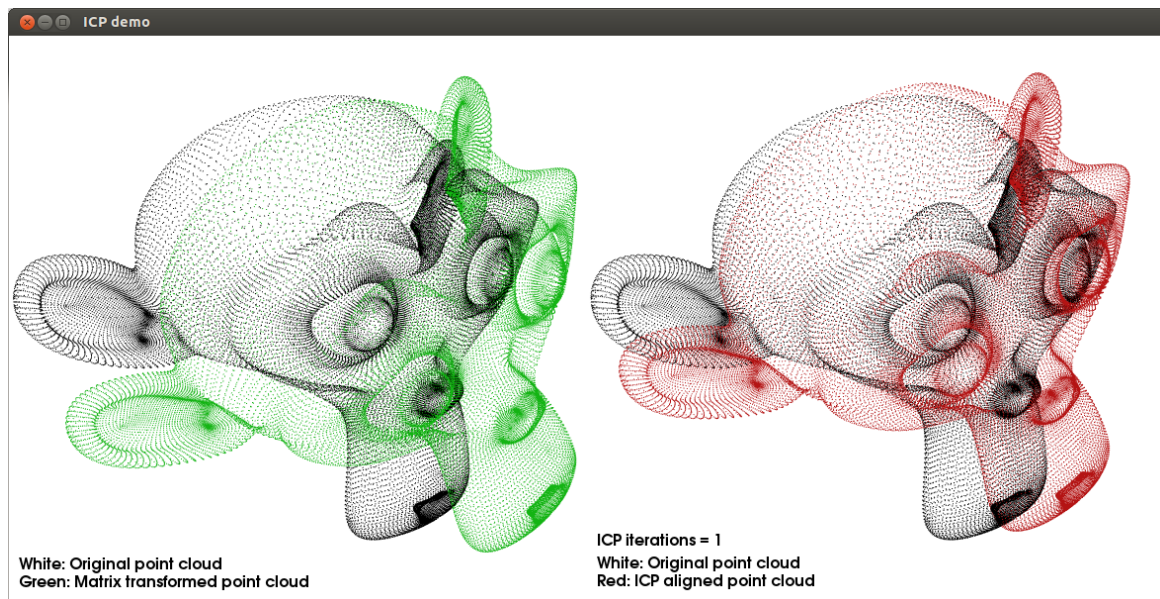


Figura 2 – Exemplo de funcionamento ICP 3D [26]

Uma vez que a transformação é estimada, o resultado pode ser usado para se obter a posição e orientação do sensor LIDAR em relação à nuvem de pontos de referência ou ao mapa. Além da estimativa de localização, o ICP também pode ser utilizado para estimar outros valores, como velocidade e aceleração, através do cálculo da diferença entre as transformações estimadas em diferentes instantes ao longo do tempo. Com os dados obtidos da transformação, Assim, o filtro de Kalman é alimentado com os dados obtidos da transformação, ou seja, a nuvem de pontos para posição, rotação, translação, velocidade, entre outros.

2.1.2 Filtro de Kalman

O filtro de Kalman, componente central do SLAM, é um algoritmo de predição e rastreamento de objetos que utiliza medições realizadas ao longo do tempo (contaminadas com ruído e outras incertezas), gerando resultados próximos aos valores reais das grandezas medidas e associadas. Esse filtro combina informações de medições anteriores, usando um modelo linear onde os erros de medição e da dinâmica do sistema são independentes, com informações do modelo dinâmico para estimar o estado atual de maneira ótima. Com base nesse modelo, o algoritmo fornece uma estimativa de estado mais precisa que os métodos anteriores, mesmo na presença de ruído.

O modelo deste filtro utiliza de medidas espaciais realizadas iterativamente para realizar suas predições, são elas posição (x, y) e orientação no plano cartesiano, velocidade de deslocamento do objeto de estudo e aceleração. Combinando essas informações é possível, assim, estimar a posição atual no plano e prever qual será a posição na próxima iteração.

A execução do algoritmo se inicia com os parâmetros assumindo quaisquer valores. A cada iteração, o algoritmo computa um valor estimado e um predito sobre o objeto em análise/observação. O algoritmo sempre converge de maneira correta, mesmo para valores atribuídos inicialmente muito diferentes da realidade. Nesse caso, ocorre que as primeiras iterações do algoritmo resultam em valores bastante distintos, gerando uma convergência mais ou menos responsiva, como demonstra a imagem (...).

O valor estimado do sistema indica o estado atual mais próximo do exato, considerando medições imprecisadas, ruídos do ambiente e atrasos de envio e recebimento de dados (lag) entre iterações. Essa estimativa leva em conta a predição da iteração anterior. Já o valor predito numa iteração é uma suposição sobre o estado do sistema na próxima iteração e que leva em conta o valor estimado da iteração atual. Assim, a ideia central do filtro de Kalman é combinar estimativas ponderadas do estado do sistema de uma iteração prévia com uma nova medição, levando-se em conta as incertezas mencionadas. O filtro, ainda, atribui os maiores pesos às informações mais confiáveis e os menores às informações menos confiáveis, permitindo que o algoritmo se adapte dinamicamente às mudanças no sistema e melhore a precisão das estimativas realizadas.

2.2 O Sistema SLAM Unidimensional

A arquitetura previamente apresentada mostra como os sensores e sistemas auxiliares se interligam com o SLAM. Já esta seção é dedicada especialmente a introdução do SLAM com o filtro de Kalman unidimensional. Porém, primeiramente são estabelecidos alguns conceitos matemáticos básicos, tais como média e variância, e alguns conceitos mais avançados, tais como distribuição multivariada e covariância, que serão úteis mais

adiante. Em seguida, as equações e transformações que definem o modelo matemático do sistema de predição unidimensional são abordadas. O objetivo é mostrar como o modelo matemático calcula as estimativas dos sistemas com base nas observações externas, no modelo de movimento e nas informações estatísticas. Por fim, o filtro de Kalman multivariado, uma extensão do modelo unidimensional que lida com múltiplas variáveis em sistemas mais complexo é apresentado.

2.2.1 Conceitos e Notações

O SLAM exige alguns conceitos fundamentais, além de uma notação apropriada, para facilitar sua compreensão e implementação. Portanto, alguns conceitos estatísticos e princípios de estimação são descritos nesta seção.

Média: estado do sistema conhecido, onde a concentração dos dados em uma distribuição é maior, dada por μ ;

Valor esperado: média ponderada de eventos aleatórios levando em conta suas probabilidades associadas, ou seja, representa o valor médio de uma variável aleatória, calculado ponderando os possíveis valores dessa variável pela probabilidade de ocorrência de cada um desses valores, dado por E ;

Variância: quantifica o grau de dispersão dos valores em um conjunto de dados em relação à média, representada por σ^2 ;

- Variância da estimativa do estado, é o grau de erro sobre uma estimativa para o valor real, representada por p
- Variância da medição, é o desvio padrão entre o valor real e o valor medido, representada por r

Covariância: diferente da variância, não se concentra na dispersão ou variabilidade de uma única variável aleatória, mas sim na relação entre duas variáveis aleatórias. Enquanto a variância mede o quanto os valores de uma única variável se afastam de sua média, a covariância avalia como duas variáveis aleatórias se movem em conjunto. Assim, a covariância fornece informações sobre como duas variáveis tendem a variar simultaneamente, aumentando ou diminuindo juntas ou se seus comportamentos são independentes;

Desvio padrão: quantifica a dispersão dos valores sobre um conjunto de dados em relação à média, e é calculado como a raiz quadrada da variância, representado por σ ;

Distribuição normal: ou distribuição gaussiana, é o valor da média mais ou menos o desvio padrão, descrita por momentos/intervalos de acordo com o desvio padrão;

Distribuição normal multivariada: descreve a distribuição conjunta de múltiplas variáveis aleatórias interdependentes em um espaço multidimensional, representando a incerteza conjunta dessas variáveis. Nesse caso, a incerteza de cada variável é considerada em conjunto com outra variáveis, modelando a propagação da incerteza e considerando suas relações e correlações para inferir como os dados estão distribuídos em torno da média numa distribuição normal para calcular intervalos de confiança em análises estatísticas.

A Figura 3 mostra como duas variáveis, x e y , estão relacionadas de acordo com o intervalo de confiança, a altura, nesse caso. Note que, quando x e y se aproximam de 0, a confiança do modelo é alta, ou seja, o grau que mede quanto a relação de valores das variáveis está próxima da realidade, tende a aumentar. Já quando x e y se afastam de 0, positivamente ou negativamente, a confiança diminui.

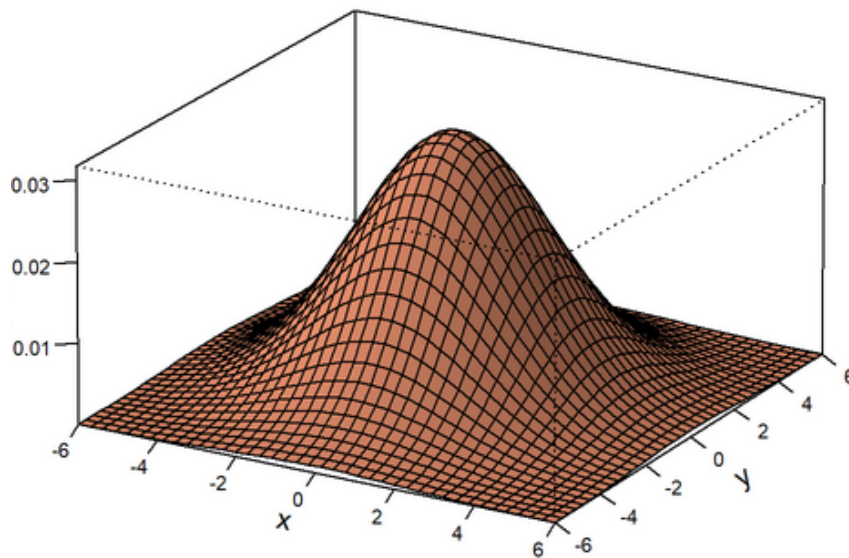


Figura 3 – Representação dos intervalos de confiança sobre uma distribuição normal gaussiana que correlaciona duas variáveis (x e y) [27].

Incerteza: mede a variabilidade ou falta de precisão associada a uma distribuição de probabilidade, representada pela dispersão dos dados em relação à média multivariada;

Variável aleatória: definida por uma função que associa valores numéricos possíveis a eventos num espaço amostral de um experimento probabilístico, e pode ser de dois tipos:

- contínua, quando assume qualquer valor dentro de um intervalo dado;
- discreta, quando o número de valores possíveis que a variável assume é finito ou infinito enumerável, ou seja, é possível listar todos os valores possíveis que a variável pode assumir.

Momento: descreve a forma e o comportamento da distribuição num determinado instante, ou seja, a média e distribuição normal num instante k . O momento pode, ainda, ser classificado em dois tipos:

- Momento bruto, calculado diretamente a partir dos dados originais ou da distribuição de probabilidade, dado pelo valor esperado da k -ésima potência da variável aleatória, $E(X^k)$;
- Momento central, calculado em relação à média da distribuição ou conjunto de dados, dado pelo valor esperado da k -ésima potência da distribuição de variáveis aleatórias em torno de sua média, $E((X - \mu_X)^k)$.

Estimativa: aproximação ou cálculo de um valor desconhecido baseado em informações disponíveis, ou seja, avalia o estado “oculto”(não conhecido) do sistema de acordo com dados provenientes de maneira interna ou externa, por exemplo dos sensores;

Acurácia/exatidão: indica o quão próxima a medição está do valor real absoluto;

Peso: importância ou influência de algo (ponderação), usado no cálculo das médias ponderadas como medição, estimativa, representado por w ;

Operações de matrizes: englobam a adição e multiplicação de matrizes e vetores, bem como sua transposição, inversão e simetria;

Autovalores: números que representam a escala de estiramento ou encolhimento de uma transformação linear representada por uma matriz;

Autovetores: vetores que não mudam de direção quando multiplicados por uma matriz ou operador linear, apenas modificando-a por um fator escalar (autovalor);

Álgebra de expectativas: refere-se ao cálculo de valores médios ponderados de eventos aleatórios, levando em consideração suas probabilidades;

Translação: movimento de um objeto em linha reta de um ponto a outro, sem alterar sua orientação, ou seja, o movimento retilíneo de um objeto de um lugar para outro.

Rotação: movimento de um objeto em torno de um ponto ou eixo, mantendo sua forma, mas mudando sua orientação;

Orientação: descreve a direção ou ângulo de um objeto em relação a um ponto de referência.

Algumas notações importantes, que serão usadas ao longo deste trabalho, também são estabelecidas a seguir:

- x representa o valor real do estado (por exemplo, peso de uma pessoa ou posição atual do robô);
- z_n representa o valor medido no tempo n em relação à uma determinada medida, podendo ser a leitura, por exemplo, da posição, orientação ou velocidade do sistema;
- $\hat{\cdot}$ representa o valor estimado, após a aplicação das equações do modelo, ou seja, após uma leitura ruidosa e dos tratamentos realizados, espera-se que este seja o valor mais próximo possível do real;
- $\hat{x}_{n,n}$ representa a estimativa de x no tempo n depois da medida de z_n ;
- $\hat{x}_{n+1,n}$ representa a predição de x em um estado futuro ($n + 1$) no tempo n ;
- α representa a precisão da medição, de 0 a 1 em escala quantitativa crescente, para posição, indicando o quão próxima a leitura dos dados está do valor real;
- β representa a precisão da medição para velocidade;
- γ representa a precisão da medição para aceleração;
- K_n representa o ganho de kalman, uma variável que controla a contribuição das estimativas e predições anteriores e das observações mais recentes para a estimativa atual do estado do sistema. O valor da variável é dado entre 0 a 1 em escala quantitativa crescente, onde o objetivo é minimizar o erro da estimativa;
- \dot{x} representa a derivada do deslocamento de x sobre o tempo, resultando na velocidade;
- \ddot{x} representa a derivada da variação de \dot{x} sobre o tempo ao quadrado, resultando na aceleração;
- $p_{n,n}$ representa a variância da estimativa no tempo n ;
- $p_{n,n-1}$ representa a variância da estimativa anterior;
- r_n representa a variância da medição atual;
- Letras minúsculas com uma seta em cima apontando para a direita referem-se a vetores, como \vec{x} ;
- Letras maiúsculas com uma seta em cima apontando para a direita referem-se a vetores referem-se a matrizes, como \vec{A} ;
- Letras minúsculas normais referem-se a escalares ou elementos vetoriais, a ;
- Letras maiúsculas normais referem-se a elementos de matriz, A ;

- $P_{n,n}$ representa a matriz de covariância que descreve a incerteza ao quadrado de uma estimativa;
- $P_{n+1,n}$ matriz de covariância que descreve a incerteza ao quadrado de uma previsão;
- R_n matriz de covariância que descreve a incerteza da medição ao quadrado;

2.2.2 Modelo de Predição Unidimensional

O modelo de predição do filtro de Kalman unidimensional é uma abordagem amplamente utilizada para estimar e prever o estado de sistemas dinâmicos em tempo real, mesmo quando esses sistemas estão sujeitos a ruídos e incertezas. Nessa seção são descritas as equações do modelo que desempenham papéis cruciais na atualização e projeção dos estados do sistema e na gestão da incerteza associada a esses estados. O conjunto de equações do modelo pode ser dividido em equações de: atualização de estado, extrapolação de estado, relação entre variâncias de medição e predição, ganho de Kalman, atualização de covariância, e predição de covariância.

As equações de atualização de estado calcula o novo estado estimado do sistema com base nas observações mais recentes e nas estimativas anteriores. O resultado dessas equações refletem como as novas informações observadas ajustam o estado estimado, levando em consideração a incerteza associada às observações. O cálculo depende de α , β e γ para determinar o quão impactante será cada leitura no decorrer do sistema. Assim, quanto mais precisas são as leituras, ou seja, α , β e γ maiores, maior será o seu impacto na dinâmica do sistema. As equações para calcular, respectivamente, os estados de posição, velocidade e aceleração são dados por

$$\begin{aligned}\hat{x}_{n,n} &= \hat{x}_{n,n-1} + \alpha * (z_n - \hat{x}_{n,n-1}) \\ \hat{\dot{x}}_{n,n} &= \hat{\dot{x}}_{n,n-1} + \beta * \frac{z_n - \hat{x}_{n,n-1}}{\Delta t} \\ \hat{\ddot{x}}_{n,n} &= \hat{\ddot{x}}_{n,n-1} + \gamma * \frac{z_n - \hat{x}_{n,n-1}}{0.5 * \Delta t^2}\end{aligned}\tag{2.1}$$

As equações de extrapolação de estado são obtidas pela estimativa calculada na iteração atual para prever a posição do sistema na próxima iteração. Essas equações de predição dos novos estados de posição, velocidade e aceleração são representados, respectivamente, por

$$\begin{aligned}\hat{x}_{n+1,n} &= \hat{x}_{n,n} + \hat{\dot{x}}_{n,n} * \Delta t + \hat{\ddot{x}}_{n,n} * \left(\frac{\Delta t^2}{2}\right) \\ \hat{\dot{x}}_{n+1,n} &= \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n} * \Delta t \\ \hat{\ddot{x}}_{n+1,n} &= \hat{\ddot{x}}_{n,n}\end{aligned}\tag{2.2}$$

A relação entre as variâncias de medição e predição é uma medida de incerteza que combina informações provenientes de duas fontes diferentes, as medições reais e as

estimativas de estado baseadas no modelo de predição do sistema. O resultado é uma relação entre o quão precisa é uma medição e o quão preciso é o modelo de predição do sistema. Dessa forma, as variâncias de ambos os dados são levadas em conta para determinar a melhor estimativa do estado atual do sistema, representada pela equação $p_{n,n} = (w_1)^2 * r_n + (w_2)^2 * p_{n,n-1}$ sendo $w_1 + w_2 = 1$.

A equação do ganho de Kalman determina o peso que deve ser atribuído às observações atuais em comparação com as estimativas anteriores ao atualizar o estado. Como o objetivo do filtro é achar uma estimativa ótima, deve-se minimizar a sua variância, resultando em

$$\begin{aligned} p_{n,n} &= (w_1)^2 * r_n + (1 - w_1)^2 * p_{n,n-1} \Rightarrow \\ \frac{dp_{n,n}}{dw_1} &= 2 * w - 1 * r_n - 2 * (1 - w_1) * p_{n,n-1} = 0 \Rightarrow \\ w_1 &= \frac{p_{n,n-1}}{p_{n,n-1} + r_n} = K_n \end{aligned} \quad (2.3)$$

Substituindo na equação de estimação do estado atual obtém-se que

$$\hat{x}_{n,n} = K_n * z_n + (1 - K_n) * \hat{x}_{n,n-1} = \hat{x}_{n,n-1} + K_n * (z_n - \hat{x}_{n,n-1}). \quad (2.4)$$

Observe que o ganho de Kalman é um balanceamento entre variâncias, da predição da estimativa anterior e da medição atual, para minimizar a variância ao calcular a estimativa do estado atual. Esse balanceamento é melhor quando comparado com os filtros α - β - γ , onde as medições são recebidas como parâmetros imutáveis, como por exemplo quando um sensor de distância possui uma precisão de 2 centímetros, indicando que o valor medido pode variar em até 2 centímetros, positivamente ou negativamente.

A atualização do valor de covariância é dada pela equação

$$p_{n,n} = (K_n)^2 * r_n + (1 - K_n)^2 * p_{n,n-1} \quad (2.5)$$

A cada iteração esse valor é atualizado e usado em outras equações, refletindo o grau de confiança que o filtro de Kalman possui sobre as estimativas. A atualização da covariância leva em consideração a precisão das medições e a dinâmica do sistema, ajustando a incerteza conforme novas iterações são realizadas.

Já para encontrar o valor de w_2 , dado por $1 - K_n$, temos que

$$1 - K_n = 1 - p_{n,n-1}/(p_{n,n-1} + r_n) = r_n/(p_{n,n-1} + r_n) \quad (2.6)$$

Substituindo, o resultado obtido para $p_{n,n}$ é dado por

$$\begin{aligned} p_{n,n} &= \left(\frac{p_{n,n-1}}{p_{n,n-1} + r_n}\right)^2 * r_n + \left(\frac{r_n}{p_{n,n-1} + r_n}\right)^2 * p_{n,n-1} = \\ \frac{p_{n,n-1} * r_n}{p_{n,n-1}} + r_n * \frac{p_{n,n-1}}{p_{n,n-1} + r_n} + \frac{r_n}{p_{n,n-1} + r_n} &= (1 - K_n) * p_{n,n-1}. \end{aligned} \quad (2.7)$$

A equação de predição da covariância tem o objetivo de reduzir erros de estimativa ao adicionar incerteza do modelo dinâmico sobre a leitura de dados, ou seja, ruídos que afetam o resultado de uma iteração, levando a uma propagação futura, que não pode ser prevista. Por exemplo, o valor de um resistor pode mudar entre as iterações devido a temperatura do ambiente. Logo, a equação representa o quão volátil é uma leitura de fato, como um intervalo de precisão, dado por

$$p_{n+1,n} = p_{n,n} + q_n. \quad (2.8)$$

De forma resumida, as equações do filtro de Kalman são:

1. Atualização de estado

$$\hat{x}_{n,n} = \hat{x}_{n,n-1} + K_n * (z_n - \hat{x}_{n,n-1})$$

2. Atualização de covariância

$$p_{n,n} = (1 - K_n) * p_{n,n-1}$$

3. Ganho de Kalman

$$K_n = \frac{p_{n,n-1}}{p_{n,n-1} + r_n}$$

4. Extrapolação de estado

$$\hat{x}_{n+1,n} = \hat{x}_{n,n} + \hat{\dot{x}}_{n,n} * \Delta t + \hat{\ddot{x}}_{n,n} * \frac{\Delta t^2}{2}$$

$$\hat{\dot{x}}_{n+1,n} = \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n} * \Delta t$$

$$\hat{\ddot{x}}_{n+1,n} = \hat{\ddot{x}}_{n,n}$$

5. Extrapolação de covariância

$$p_{n+1,n}^v = p_{n,n}^v + p_{n,n}^a * \Delta t$$

$$p_{n+1,n}^a = p_{n,n}^a$$

O diagrama da Figura 4 ilustra o funcionamento e as etapas do filtro de Kalman.

2.2.3 Aplicação do filtro de Kalman unidimensional

Esta subseção apresenta um exemplo prático para ilustrar a aplicação do filtro de Kalman unidimensional. Suponha um cenário simples onde se deseja estimar a distância de um carro em relação a um radar de velocidade. O carro se move em linha reta e já ultrapassou o radar, porém não se sabe sua velocidade. O sensor que mede a distância entre o carro e o radar z_n realiza uma leitura a cada segundo, e a cada leitura é realizada uma iteração do filtro, sendo que algumas medições podem conter algum ruído/erro aleatório. O objetivo é prever a localização do carro, da maneira mais precisa possível, utilizando o filtro de Kalman unidimensional, ao invés de apenas obter os dados de leitura do radar.

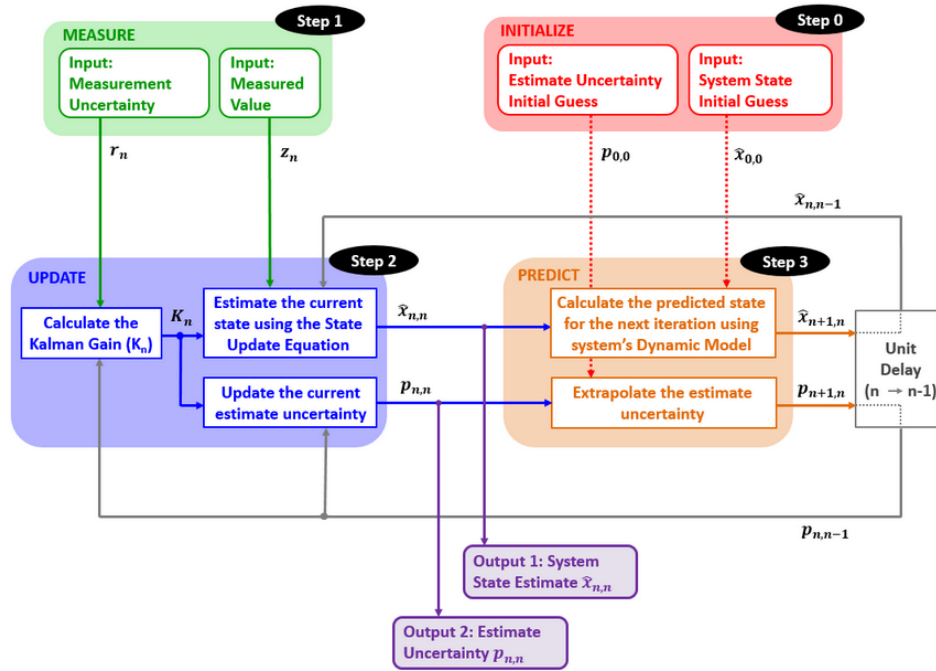


Figura 4 – Etapas do filtro de Kalman [20].

Iteração 0 (Inicialização)

Alguns parâmetros iniciais devem ser definidos para a execução do algoritmo de predição e estimação do filtro de Kalman. Primeiramente, a distância entre carro e radar ($\hat{x}_{0,0}$) é definida, de forma aleatória, em 10 metros. Como o palpite é aleatório e impreciso, os parâmetros são inicializados da seguinte forma: (i) a covariância $p_{n,n}$ é iniciada com um valor alto, pois não se sabe de fato o estado atual do carro no primeiro instante. Logo, $p_{0,0} = 10.000$, indicando uma alta imprecisão no palpite inicial; (ii) a incerteza q , que varia de 0 a 1, é inicializada com $q = 0,15$, um valor relativamente alto se comparado com outros modelos quando se tem alguma base para o chute inicial; e (iii) o baixo ruído de medição $r = 0,01$, que representa a porcentagem de quão distante estão as medições em relação aos valores reais, indica que as medições são mais precisas e confiáveis.

Vale também ressaltar que a incerteza q pode ser inicializada com um valor muito alto, por exemplo, 0,9. A diferença está no número de iterações necessárias para que o modelo se alinhe com as medições e a posição real do objeto de estudo [28]. Além disso, a velocidade não é definida no modelo do exemplo não fornecendo nenhum embasamento para o palpite inicial. Já o parâmetro de ruído é, principalmente, impactado pela precisão do equipamento utilizado. Caso o equipamento seja muito ruidoso o modelo terá, conseqüentemente, uma menor precisão.

A predição da posição na próxima iteração assume o valor da estimativa do estado atual, já que existe apenas um valor de medição (posição) para se usar de referência no sistema descrito. Portanto, a predição da posição do estado do sistema na próxima

iteração é obtida pela estimativa realizada (chute inicial) na iteração atual. Logo, temos que $\hat{x}_{1,0} = \hat{x}_{0,0}$. A predição da variância da próxima iteração é obtida pela estimativa da variância do estado atual. Assim obtemos, $p_{1,0} = p_{0,0} + q = 10.000,15$.

Iteração 1

Nessa iteração o algoritmo usa a predição feita na iteração anterior sobre o estado $x_{1,0}$ para estimar o estado atual $x_{1,1}$ e prever a próxima posição do carro na estrada (próximo estado) $x_{2,1}$.

- Medição (z_n) = 49,732 m
- Estimativas do estado atual:

$$K_1 = \frac{10.000,15}{10.000,15+0.01} = 0,999999$$

A porcentagem alta no ganho de Kalman nessa iteração indica alta confiabilidade nos dados medidos e baixa confiabilidade nos dados estimados/preditos.

$$\hat{x}_{1,1} = 10 + 0,999999 * (49,732 - 10) = 49,732m$$

O cálculo da estimativa do estado atual $\hat{x}_{1,1}$ resultou no valor igual ao medido devido a porcentagem do ganho de Kalman estar muito alto, próximo de 1, priorizando o valor medido ao invés do estimado.

$$p_{1,1} = (1 - 0,999999) * 10.000,15 = 0,01.$$

A covariância do estado atual $p_{1,1}$ bla bla bla

- Predição do próximo estado:

$$\hat{x}_{2,1} = \hat{x}_{1,1} = 49,732m$$

A extrapolação de estado $\hat{x}_{2,1}$ é obtida pela forma reduzida, já que o exemplo considera apenas o parâmetro de posição.

$$p_{2,1} = 0,01 + 0,15 = 0,16$$

A extrapolação de covariância $p_{2,1}$ também é obtida pela forma reduzida, usando a equação 2.8.

Iteração 2

Na iteração 2 o processo se repete para estimar a posição do estado atual com a predição realizada na iteração 1, juntamente com a medição atual, podendo então realizar a predição do próximo estado. Nas próximas iterações o processo continua se repetindo com a realização das medições e estimativas.

- Medição (z_n) = 59,781 m

- Estimativas do estado atual:

$$K_2 = \frac{0,16}{0,16+0,01} = 0,9412$$

$$\hat{x}_{2,2} = 49,732 + 0,9412 * (59,781 - 49,732) = 59,190m$$

$$p_{2,2} = (1 - 0,9412) * 0,16 = 0,0094$$

- Predição do próximo estado:

$$\hat{x}_{3,2} = \hat{x}_{2,2} = 59,190m$$

$$p_{3,2} = 0,0094 + 0,15 = 0,1594$$

Iteração 3

- Medição (z_n) = 68,545 m

- Estimativas do estado atual:

$$K_3 = \frac{0,1594}{0,1594+0,01} = 0,941$$

$$\hat{x}_{3,3} = 59,190 + 0,941 * (68,545 - 59,190) = 67,993m$$

$$p_{3,3} = (1 - 0,941) * 0,1594 = 0,0094$$

- Predição do próximo estado:

$$\hat{x}_{4,3} = \hat{x}_{3,3} = 67,993m$$

$$p_{4,3} = 0,0094 + 0,15 = 0,1594$$

⋮

Iteração 12

- Medição (z_n) = 161,131 m

- Estimativas do estado atual:

$$K_{12} = \frac{0,1594}{0,1594+0,01} = 0,941$$

$$\hat{x}_{12,12} = 150,170 + 0,941 * (161,131 - 150,170) = 160,484m$$

$$p_{12,12} = (1 - 0,941) * 0,1594 = 0,0094$$

- Predição:

$$\hat{x}_{13,12} = \hat{x}_{12,12} = 160,484m$$

$$p_{13,12} = 0,0094 + 0,15 = 0,1594$$

Na décima segunda iteração é possível observar que o valor da predição calculado pelo algoritmo do filtro de Kalman está muito próximo do valor real medido. Dessa forma, as iterações podem continuar ocorrendo mesmo que o carro esteja fora do alcance do radar e não seja mais possível obter a distância do carro z_n . Nesse caso, a medição é substituída pela própria predição do estado anterior, até que apareça um outro radar mais adiante que permita, novamente, a leitura da distância z_n .

Resultados

A Figura 5 apresenta os resultados da aplicação do filtro de Kalman no cenário descrito. Percebe-se que, de acordo com os parâmetros definidos na inicialização do filtro, os dados estimados tendem a permanecer próximos dos medidos e que todos os valores estimados permaneceram dentro do intervalo de confiança de 95%. Nota-se também que, apesar de se ter a leitura real dos dados com uma precisão de 1% de erro ($r = 0,01$), o resultado da estimativa do filtro de Kalman é, ainda, mais preciso do que essa medição, resultando num valor mais próximo do real estado do objeto de estudo. Esse fato tende a ser mais aparente quando a precisão do sensor diminui.

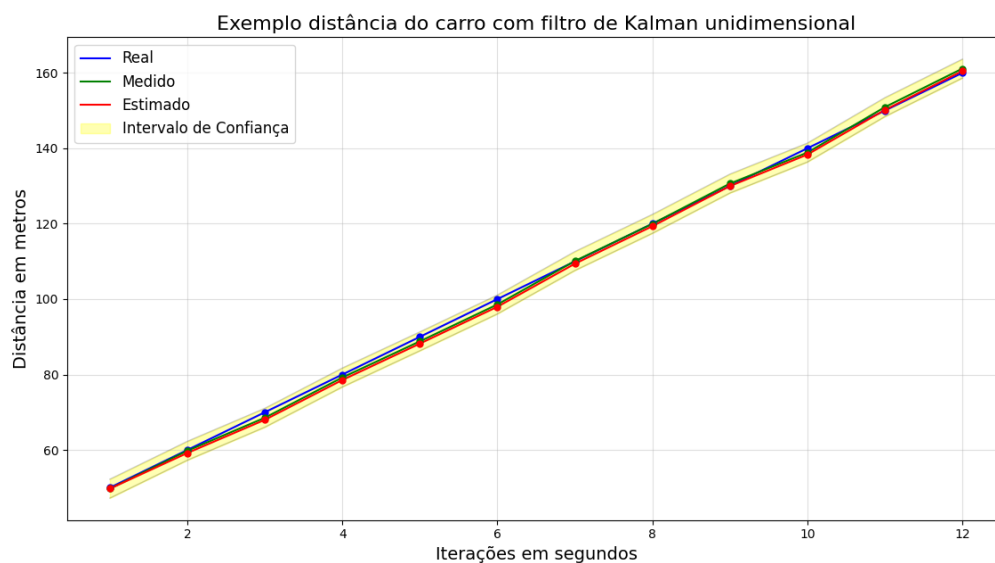


Figura 5 – Resultados do exemplo prático com filtro de Kalman multidimensional

2.3 O Sistema SLAM Multivariado

O filtro de Kalman multivariado estende o modelo unidimensional para resolver problemas em SLAM que possuam duas ou mais dimensões. Em ambientes multidimensionais, vetores e matrizes são utilizados para representar as dimensões do ambiente analisado. Num ambiente tridimensional, por exemplo, o vetor de estado $[x, y, z]$ descreve a

posição de um objeto no espaço 3D. Assim, o vetor de estado que descreve a posição e velocidade de um objeto no espaço é dado por $[x, y, z, \dot{x}, \dot{y}, \dot{z}]$, enquanto que o vetor de estado que descreve a posição, velocidade e aceleração de um objeto no espaço é representado por $[x, y, z, \dot{x}, \dot{y}, \dot{z}, \ddot{x}, \ddot{y}, \ddot{z}]$.

O objetivo é então adaptar o filtro de Kalman, originalmente desenvolvido para sistemas unidimensionais, de modo a lidar com sistemas mais complexos que operam num espaço multidimensional. Neste cenário, o modelo adaptado deve ser capaz de estimar e prever estados do sistema utilizando das mesmas fórmulas e teorias já apresentadas anteriormente no modelo unidimensional, de acordo com a quantidade de dimensões. Assim, as equações do modelo unidimensional, apresentadas na Subseção 2.2.2, devem ser adaptadas para um modelo multivariado.

Os princípios fundamentais do filtro de Kalman, tais como a atualização do estado com base em observações, a atualização da covariância e a previsão do próximo estado, permanecem os mesmos, no entanto, a representação de matrizes é introduzida para que diversas dimensões sejam modeladas. As equações são então aplicadas sobre as matrizes que modelam o espaço multidimensional, correlacionando suas variáveis num único modelo e relacionando suas derivações e variâncias. As equações são calculadas, basicamente, da mesma maneira que no modelo unidimensional, porém uma vez para cada plano bidimensional e interligadas de acordo com o cálculo da covariância e da distribuição normal multivariada.

Na adaptação do cálculo da covariância (Equação 2.8), por exemplo, as duas principais mudanças são: (a) a inserção da matriz de covariância, que captura as relações entre todas as variáveis de estado, ou seja, considera covariâncias cruzadas entre as variáveis, refletindo como essas covariâncias influenciam umas às outras; e (b) a utilização de modelagem multivariada, que envolve a consideração das influências mútuas e das interações entre as variáveis de estado em sistemas multidimensionais. Assim, o filtro de Kalman leva em consideração como as variáveis de estado estão relacionadas e como suas mudanças afetam umas às outras. A principal vantagem desse modelo é sua capacidade de lidar com sistemas mais complexos, onde as variáveis de estado estão interligadas e se influenciam. Essas e outras mudanças necessárias nas formulações das equações são abordadas na Subseção 2.3.1.

Nas próximas subseções são apresentados o modelo de predição bidimensional, juntamente com suas equações adaptadas, e uma transformação de um modelo tridimensional em bidimensional, para que este último seja aplicado em ambientes reais tridimensionais.

2.3.1 Modelo de predição Bidimensional

O modelo de predição bidimensional é uma instanciação do modelo multivariado, extensível para qualquer dimensão. Da mesma forma que o modelo unidimensional, o

modelo bidimensional segue as cinco principais das equações do filtro de Kalman unidimensional. Esta seção mostra como o modelo é definido de acordo com as medições sobre cada dimensão e como relacioná-las.

No modelo bidimensional temos duas variáveis, x e y , e os parâmetros associados de posição, velocidade e aceleração representados num único vetor da forma $\hat{\mathbf{w}} = [x, y, \dot{x}, \dot{y}, \ddot{x}, \ddot{y}]$. Assim, o cálculo da extrapolação de estado (Equação 2.2) adaptada para o ambiente bidimensional é dado pela Equação 2.9.

$$\hat{\mathbf{w}}_{n+1,n} = \mathbf{F} * \hat{\mathbf{w}}_{n,n} + \mathbf{Q} \quad (2.9)$$

Note que a predição da próxima iteração é obtida pela multiplicação da matriz \mathbf{F} de transição de estado [20] pelo vetor de estimativas do estado atual do sistema e somada com a matriz \mathbf{Q} de variâncias do objeto de estudo.

$$\begin{bmatrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{\dot{x}}_{n+1,n} \\ \hat{\dot{y}}_{n+1,n} \\ \hat{\ddot{x}}_{n+1,n} \\ \hat{\ddot{y}}_{n+1,n} \end{bmatrix} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{\dot{x}}_{n,n} \\ \hat{\dot{y}}_{n,n} \\ \hat{\ddot{x}}_{n,n} \\ \hat{\ddot{y}}_{n,n} \end{bmatrix} + \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix}$$

Logo, a predição da próxima iteração, $\hat{\mathbf{w}}_{n+1,n}$, usando as estimativas do estado atual do sistema, $\hat{\mathbf{w}}_{n,n}$, é obtida pelo sistema de equações dado abaixo, com base na Equação 2.9.

$$\hat{\mathbf{w}}_{n+1,n} = \begin{bmatrix} \hat{x}_{n+1,n} \\ \hat{y}_{n+1,n} \\ \hat{\dot{x}}_{n+1,n} \\ \hat{\dot{y}}_{n+1,n} \\ \hat{\ddot{x}}_{n+1,n} \\ \hat{\ddot{y}}_{n+1,n} \end{bmatrix} = \begin{cases} \hat{x}_{n,n} + \hat{\dot{x}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{x}}_{n,n}\Delta t^2 + q_1 \\ \hat{y}_{n,n} + \hat{\dot{y}}_{n,n}\Delta t + \frac{1}{2}\hat{\ddot{y}}_{n,n}\Delta t^2 + q_2 \\ \hat{\dot{x}}_{n,n} + \hat{\ddot{x}}_{n,n}\Delta t + q_3 \\ \hat{\dot{y}}_{n,n} + \hat{\ddot{y}}_{n,n}\Delta t + q_4 \\ \hat{\ddot{x}}_{n,n} + q_5 \\ \hat{\ddot{y}}_{n,n} + q_6 \end{cases}$$

Note que a matriz de transição de estado, \mathbf{F} , o vetor de estimativas do estado atual, $\hat{\mathbf{w}}_{n,n}$, e a matriz de covariância de ruído do processo, \mathbf{Q} , são dados como segue.

$$\mathbf{F} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} \quad \hat{\mathbf{w}}_{n,n} = \begin{bmatrix} \hat{x}_{n,n} \\ \hat{y}_{n,n} \\ \hat{\dot{x}}_{n,n} \\ \hat{\dot{y}}_{n,n} \\ \hat{\ddot{x}}_{n,n} \\ \hat{\ddot{y}}_{n,n} \end{bmatrix} \quad \mathbf{Q} = \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix}$$

Vale ressaltar que os valores q_1, q_2, \dots da diagonal principal da matriz \mathbf{Q} são as variâncias de cada variável de estado do objeto de estudo (posição, velocidade, etc) em

um exemplo onde ruído de processo é independente, ou seja, sem relação ou correlação entre as variáveis de estado. Cada elemento do vetor resultante da extrapolação de estado é então somado com seu respectivo valor de ruído da medição.

Porém, assumindo um modelo discreto onde o ruído é volátil e, portanto, varia de um instante para outro de forma constante, a matriz Q de um modelo bidimensional pode ser simplificada. Essa simplificação é obtida pela propriedade de simetria da covariância $COV(x, \dot{x}) = COV(\dot{x}, x)$ e assumindo um exemplo em que as variáveis de estado estão correlacionadas, então, a matriz Q é dada como segue.

$$Q = \begin{bmatrix} V(x) & COV(x, \dot{x}) \\ COV(\dot{x}, x) & V(\dot{x}) \end{bmatrix}$$

Além disso, como a variância é definida pela média dos quadrados das diferenças entre os valores da variável e sua média, e todo o valor esperado de uma constante não muda, a média se mantém constante, ou seja, a média permanece a mesma, independente das probabilidades e variâncias. Assim, seja z uma variável qualquer, sua variância é obtida do valor esperado E e pela média de z e μ_z .

$$\begin{aligned} V(z) &= E((z - \mu_z)^2) = E(z^2 - 2z\mu_z + \mu_z^2) = E(z^2) + E(-2z\mu_z) + E(\mu_z^2) \\ E(-2z\mu_z) &= -2\mu_z * E(z) = -2\mu_z^2 \\ E(\mu_z^2) &= \mu_z^2 \\ V(z) &= E(z^2) - \mu_z^2 \end{aligned} \quad (2.10)$$

O valor esperado de uma variável A é calculado pela fórmula

$$E(A) = \sum_{i=1}^{\infty} a_i P(a_i),$$

onde a_i é um valor específico que A pode assumir e $P(a_i)$ é probabilidade de A assumir o valor a_i . Os valores das variâncias de posição (x) e velocidade (\dot{x}) são obtidos pelas Equações 2.11 e 2.12, respectivamente, e o valor da covariância é obtido pela Equação 2.13.

$$\begin{aligned} V(x) &= E(x^2) - \mu_x^2 = E\left(\left(\frac{1}{2}\ddot{x}\Delta t^2\right)^2\right) - \left(\frac{1}{2}\mu_{\ddot{x}}\Delta t^2\right)^2 = \\ &= \frac{\Delta t^2}{4}(E(\ddot{x}^2) - \mu_{\ddot{x}}^2) = \frac{\Delta t^2}{4}\sigma_{\ddot{x}}^2 \end{aligned} \quad (2.11)$$

$$\begin{aligned} V(\dot{x}) &= E(\dot{x}^2) - \mu_{\dot{x}}^2 = E(\dot{x}^2) - \mu_{\dot{x}}^2 = \\ E((\ddot{x}\Delta t)^2) - (\mu_{\ddot{x}}\Delta t)^2 &= \Delta t^2(E(\ddot{x}^2) - \mu_{\ddot{x}}^2) = \Delta t^2\sigma_{\ddot{x}}^2 \end{aligned} \quad (2.12)$$

$$\begin{aligned} COV(x, v) &= COV(v, x) = E(xv) - \mu_x\mu_v = \\ E\left(\frac{1}{2}\ddot{x}\Delta t^2\ddot{x}\Delta t\right) - \left(\frac{1}{2}\mu_{\ddot{x}}\Delta t^2\mu_{\ddot{x}}\Delta t\right) &= \frac{\Delta t^3}{2}(E(\ddot{x}^2) - \mu_{\ddot{x}}^2) = \frac{\Delta t^3}{2}\sigma_{\ddot{x}}^2 \end{aligned} \quad (2.13)$$

Portanto, a matriz da covariância de ruído Q do processo é definida por

$$Q = \begin{bmatrix} \frac{\Delta t^2}{4} & \frac{\Delta t^3}{2} \\ \frac{\Delta t^3}{2} & \Delta t^2 \end{bmatrix} * \sigma_{\ddot{x}}^2,$$

onde $\sigma_{\ddot{x}}^2$ é a variância de aceleração, determinada com base no conhecimento do sistema dinâmico modelado, dados experimentais ou observações do sistema. Nota-se que tanto a variância de posição e velocidade quanto a covariância entre esses dois parâmetros são obtidas em função da variância de aceleração, através de derivações da álgebra de probabilidade [20].

No cálculo da extrapolação de covariância para o modelo bidimensional é necessário adaptar a Equação 2.8 do modelo unidimensional. Essa adaptação depende da matriz de ruído Q e da matriz de transição de estado F . O resultado é a Equação 2.14 que retorna a matriz de covariância da incerteza quadrática de uma predição (próximo estado) $P_{n+1,n}$, usando a matriz de covariância da incerteza quadrática (variância) de uma estimativa (estado atual) $P_{n,n}$, a matriz de transição de estado F e sua transposta F^T , e a matriz de covariância de ruído do processo Q .

$$P_{n+1,n} = F * P_{n,n} * F^T + Q \quad (2.14)$$

A covariância da incerteza quadrática de uma estimativa (estado atual) é dada pela matriz $P_{n,n}$.

$$P_{n,n} = \begin{bmatrix} p_x & p_{x\dot{x}} & p_{x\ddot{x}} & 0 & 0 & 0 \\ p_{\dot{x}x} & p_{\dot{x}} & p_{\dot{x}\ddot{x}} & 0 & 0 & 0 \\ p_{\ddot{x}x} & p_{\ddot{x}\dot{x}} & p_{\ddot{x}} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_y & p_{y\dot{y}} & p_{y\ddot{y}} \\ 0 & 0 & 0 & p_{\dot{y}y} & p_{\dot{y}} & p_{\dot{y}\ddot{y}} \\ 0 & 0 & 0 & p_{\ddot{y}y} & p_{\ddot{y}\dot{y}} & p_{\ddot{y}} \end{bmatrix}$$

A diagonal principal da matriz representa as variâncias de posição, velocidade e aceleração estimadas, de cada variável, como por exemplo p_x , $p_{\dot{x}}$ e $p_{\ddot{x}}$, respectivamente, para a variável x . O mesmo ocorre para a variável y . Já os elementos fora da diagonal principal são as covariâncias, obtidas com o filtro de Kalman ao longo das iterações. O exemplo prático da Subsecção 2.3.2 mostra que a matriz $P_{n,n}$ é inicializada apenas com os elementos da diagonal principal, enquanto o restante dos elementos são iniciados com o valor 0. Conforme as iterações do filtro são executadas, os valores das covariâncias são gerados. Esses valores das covariâncias representam a relação entre as variáveis de estado, portanto, assim como $COV(x, \dot{x}) = COV(\dot{x}, x)$, $p_{x\dot{x}} = p_{\dot{x}x}$, $p_{\dot{x}\ddot{x}} = p_{\ddot{x}\dot{x}}$, etc, na matriz $P_{n,n}$.

Logo, a matriz resultante $\mathbf{P}_{n+1,n}$ do próximo estado é obtida como segue.

$$\mathbf{P}_{n+1,n} = \begin{bmatrix} 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 & 0 \\ 0 & 1 & 0 & \Delta t & 0 & 0.5\Delta t^2 \\ 0 & 0 & 1 & 0 & \Delta t & 0 \\ 0 & 0 & 0 & 1 & 0 & \Delta t \\ 0 & 0 & 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \begin{bmatrix} p_x & p_{x\dot{x}} & p_{x\ddot{x}} & 0 & 0 & 0 \\ p_{\dot{x}x} & p_{\dot{x}} & p_{\ddot{x}} & 0 & 0 & 0 \\ p_{\ddot{x}x} & p_{\ddot{x}\dot{x}} & p_{\ddot{x}} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_y & p_{y\dot{y}} & p_{y\ddot{y}} \\ 0 & 0 & 0 & p_{\dot{y}y} & p_{\dot{y}} & p_{\ddot{y}} \\ 0 & 0 & 0 & p_{\ddot{y}y} & p_{\ddot{y}\dot{y}} & p_{\ddot{y}} \end{bmatrix}$$

$$* \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 0 & 0 \\ \Delta t & 0 & 1 & 0 & 0 & 0 \\ 0 & \Delta t & 0 & 1 & 0 & 0 \\ 0.5\Delta t^2 & \Delta t & 0 & 0 & 1 & 0 \\ 0 & 0.5\Delta t^2 & 0 & \Delta t & 0 & 1 \end{bmatrix} + \begin{bmatrix} q_1 & 0 & 0 & 0 & 0 & 0 \\ 0 & q_2 & 0 & 0 & 0 & 0 \\ 0 & 0 & q_3 & 0 & 0 & 0 \\ 0 & 0 & 0 & q_4 & 0 & 0 \\ 0 & 0 & 0 & 0 & q_5 & 0 \\ 0 & 0 & 0 & 0 & 0 & q_6 \end{bmatrix}$$

Os valores das variâncias na diagonal principal da matriz $\mathbf{P}_{n+1,n}$ são, geralmente, inicializados com valores altos em comparação com o erro da medição, por se tratar de um palpite na inicialização do filtro.

Já a multiplicação pela matriz \mathbf{F}^T , a transposta de \mathbf{F} , garante a coerência matemática do filtro, ou seja, mantém a orientação e o sentido do objeto de estudo representado na forma matricial. Vale lembrar que a matriz \mathbf{F} de transição é necessária para se obter o sistema de equações que resulta no cálculo da extrapolação de estado. A matriz \mathbf{F}^T também reflete o relacionamento entre as variáveis de estado em diferentes instantes de tempo. Na equação de covariância, por exemplo, as variáveis x e y estão inversamente relacionadas por uma multiplicação. Logo, o aumento da variável x implica na diminuição de y para que a proporção entre tais variáveis se mantenha. Já numa divisão, a relação se mantém diretamente proporcional entre as variáveis, pois o aumento de x também implica no aumento de y . Observe também que as matrizes de covariância são sensíveis à orientação e escala dos sistemas de coordenadas em que estão definidas. Por isso, a multiplicação pela matriz transposta permite uma transformação apropriada da matriz de covariância de incerteza do estado atual para o próximo estado.

Assim a simulação de movimento do objeto de estudo no ambiente se dá, a cada iteração do filtro, através de uma rotação, um giro do objeto sobre seu centro de acordo com um certo ângulo, e uma translação, o movimento do objeto de um ponto a outro mantendo sua orientação. Esses movimentos de rotação e translação são realizados sobre o mapa do ambiente em relação a uma leitura anterior e a atual, enquanto que a multiplicação pela matriz \mathbf{F}^T resulta numa matriz de rotação e translação 2D, conforme os cálculos obtidos da extrapolação de covariância entre as iterações. A Subseção 2.3.2 apresenta um exemplo que ilustra o papel dessas multiplicações de forma prática.

Já a equação de atualização de estado do modelo é adaptada para o sistema bidimensional com a inserção da matriz de observação \mathbf{H} . Essa matriz tem a função de converter o estado do sistema (input) em medições (output) usando transformações lineares. O cálculo de atualização de estado (Equação 2.4) é obtido para o modelo bidimensional através da inclusão da matriz de observação na fórmula. A atualização de estado $\hat{\mathbf{w}}_{n,n}$ no modelo bidimensional é então definido pela Equação 2.15.

$$\hat{\mathbf{w}}_{n,n} = \hat{\mathbf{w}}_{n,n-1} + \mathbf{K}_n * (\mathbf{z}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1}) \quad (2.15)$$

A matriz \mathbf{z}_n representa a medição do sensor sobre o ambiente, retornando sua posição medida na iteração atual sobre cada dimensão. No caso de um sensor que mede apenas a posição, a matriz \mathbf{z}_n tem um tamanho de 2×1 , com os valores da posição nas coordenadas x e y , respectivamente.

Já $\hat{\mathbf{w}}_{n,n-1}$, de dimensões 6×1 , representa os valores das variáveis de estado preditos na iteração anterior relativos a cada dimensão, posição, velocidade e aceleração em x e y , como já mostrado anteriormente no desenvolvimento da Equação 2.9. Cada valor da matriz \mathbf{K}_n , do ganho de Kalman no modelo bidimensional, representa o valor do ganho de Kalman resultante da associação dos valores de $\hat{\mathbf{w}}_{n,n-1}$ e \mathbf{z}_n , pois \mathbf{K}_n recebe a multiplicação de $\mathbf{z}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1}$, resultando em uma 2×1 , e a soma de $\hat{\mathbf{w}}_{n,n-1}$, de dimensões 6×1 , ou seja, tendo um tamanho de 6×2 por ter seis variáveis de estado diferentes e duas medições. As somas, subtrações e multiplicações seguem as mesmas para matrizes de dimensões distintas, mudando apenas o tamanho de cada uma delas, sendo que \mathbf{K}_n deve seguir as dimensões em linha e coluna, respectivamente, pelas diferentes variáveis de ambiente em $\hat{\mathbf{w}}_{n,n-1}$ e pela quantidade de medições (posição, velocidade e aceleração) em \mathbf{z}_n .

A matriz de observação \mathbf{H} , de dimensões $n \times m$, representa os valores mensuráveis do ambiente, onde n é o número de variáveis de estado mensuráveis do sistema, de acordo com \mathbf{z}_n , e m é o número total de variáveis de estado, de acordo com $\hat{\mathbf{w}}_{n,n-1}$. No caso do exemplo anterior a matriz \mathbf{H} tem dimensões 6×2 . Como o vetor de estado $\hat{\mathbf{w}}_{n,n-1}$ tem seis variáveis de estado, mas apenas duas são consideradas mensuráveis no vetor de medições \mathbf{z}_n (a primeira, posição em x , e a quarta, posição em y), a matriz de observação \mathbf{H} tem um tamanho de dimensões mensuráveis em $\mathbf{z}_n \times$ variáveis de estado em $\hat{\mathbf{w}}_{n,n-1}$, resultando em uma 2×6 da forma:

$$\mathbf{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

Cada dimensão mensurável tem o valor 1 no elemento da diagonal principal $H_{[n,m]}$. O cálculo de $(\mathbf{z}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1})$, numa nova iteração nas etapas do filtro, adiciona informação apenas das dimensões que de fato são mensuradas por um sensor no ambiente. Assim, a matriz resultante deste cálculo se reduz a

$$(\mathbf{z}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1}) = \begin{bmatrix} (\mathbf{z}_1 - \hat{\mathbf{w}}_1) \\ (\mathbf{z}_4 - \hat{\mathbf{w}}_4) \end{bmatrix}$$

Logo, a função básica da matriz de observação é excluir dimensões não mensuráveis nas iterações.

A equação de atualização de covariância e sua derivação vêm de duas outras equações, da atualização de estado (Equação 2.1), e da covariância estimada (Equação 2.5). A atualização de estado do modelo bidimensional é então definida pela Equação 2.16 através da matriz de observação (H) e da matriz de erro da medição \mathbf{v}_n , que possui a mesma dimensão da matriz Q na Equação 2.9, e que representa o quão distante, positivamente ou negativamente, a medição está do valor real.

$$\hat{\mathbf{w}}_{n,n} = \hat{\mathbf{w}}_{n,n-1} + \mathbf{K}_n * (\mathbf{H}\mathbf{w}_n + \mathbf{v}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1}) \quad (2.16)$$

Note que a matriz \mathbf{w}_n na fórmula, de dimensões iguais à $\hat{\mathbf{w}}_{n,n}$, representa o estado real do sistema, ou seja, onde de fato o objeto de estudo se encontra na realidade, assim como \mathbf{v}_n , também de dimensões iguais à $\hat{\mathbf{w}}_{n,n}$, representa o erro de medição do sensor em relação ao valor real. Como citado em [20], os termos \mathbf{w}_n e \mathbf{v}_n normalmente não aparecem nos cálculos, com exceção de exemplos demonstrativos, uma vez que são desconhecidos, pois caso fosse possível ter conhecimento do verdadeiro estado do objeto ou o valor exato do erro de medição, não seria necessário o desenvolvimento do filtro de Kalman muito menos do SLAM, portanto são utilizados para ser possível realizar as derivações com as variáveis do sistema que se tem conhecimento.

O erro de estimativa é dado pela fórmula $e_n = \mathbf{w}_n - \hat{\mathbf{w}}_{n,n}$. Usando a Equação 2.16 obtemos a seguinte derivação.

$$\begin{aligned} e_n = \mathbf{w}_n - \hat{\mathbf{w}}_{n,n} &= \mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1} - \mathbf{K}_n * (\mathbf{H}\mathbf{w}_n + \mathbf{v}_n - \mathbf{H} * \hat{\mathbf{w}}_{n,n-1}) = \\ &= \mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1} - \mathbf{K}_n * \mathbf{H} * (\mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1}) - \mathbf{K}_n * \mathbf{v}_n = \\ &= (\mathbf{I} - \mathbf{K}_n\mathbf{H})(\mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1}) - \mathbf{K}_n\mathbf{v}_n \end{aligned}$$

Logo, o erro de estimativa e_n se reduz a Equação 2.17.

$$e_n = (\mathbf{I} - \mathbf{K}_n\mathbf{H})(\mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1}) - \mathbf{K}_n\mathbf{v}_n \quad (2.17)$$

Note que a matriz identidade, I , de dimensões iguais as de $P_{n,n}$, tem a função de não descartar os dados mensurados em $\mathbf{w}_n - \hat{\mathbf{w}}_{n,n-1}$. Como a multiplicação pela matriz H ignora determinadas dimensões, a subtração de I por $\mathbf{K}_n\mathbf{H}$ impede que as medições sejam desconsideradas, obtendo valores maiores que 0 na diagonal principal.

Na sequência, a covariância estimada é aplicada para se obter a covariância $P_{n,n}$

usando o erro estimativa.

$$\begin{aligned}
\mathbf{P}_{n,n} &= E(e_n * e_n^T) = \\
&E(((\mathbf{I} - \mathbf{K}_n \mathbf{H})(w_n - \hat{w}_{n,n-1}) - \mathbf{K}_n \mathbf{v}_n)((\mathbf{I} - \mathbf{K}_n \mathbf{H})(w_n - \mathbf{w}_{n,n-1}) - (\mathbf{K}_n \mathbf{v}_n)^T)) \\
&= E(((\mathbf{I} - \mathbf{K}_n \mathbf{H})(w_n - \hat{w}_{n,n-1}) - \mathbf{K}_n \mathbf{v}_n)((w_n - \hat{w}_{n,n-1})^T (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T - (\mathbf{K}_n \mathbf{v}_n)^T)) \\
&= (\mathbf{I} - \mathbf{K}_n \mathbf{H}) \underbrace{E((w_n - \hat{w}_{n,n-1})(w_n - \hat{w}_{n,n-1})^T)}_1 (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T \\
&\quad - \underbrace{E((\mathbf{I} - \mathbf{K}_n \mathbf{H})(w_n - \hat{w}_{n,n-1})(\mathbf{K}_n \mathbf{v}_n)^T)}_2 - \underbrace{E(\mathbf{K}_n \mathbf{v}_n (w_n - \hat{w}_{n,n-1})^T (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T)}_3 \\
&\quad + \mathbf{K}_n * \underbrace{E(\mathbf{v}_n \mathbf{v}_n^T)}_4 * \mathbf{K}_n^T
\end{aligned}$$

Como $w_n - \hat{w}_{n,n-1}$ é o erro da previsão de uma iteração prévia e não está relacionado com o ruído da medição atual, o valor esperado do produto de duas variáveis não relacionadas, tais como $E(w_n - \hat{w}_{n,n-1})$ e $E((w_n - \hat{w}_{n,n-1})^T)$, é 0. Portanto, as derivações a partir de 1, 2, 3 e 4, do cálculo de $\mathbf{P}_{n,n}$, são apresentados a seguir.

$$1 \left\{ E((w_n - \mathbf{w}_{n,n-1})(w_n - \mathbf{w}_{n,n-1})^T) = \mathbf{P}_{n,n-1} \right.$$

$$2 \left\{ \begin{aligned} &E((\mathbf{I} - \mathbf{K}_n \mathbf{H})(w_n - \hat{w}_{n,n-1})(\mathbf{K}_n \mathbf{v}_n)^T) = \\ &E(\mathbf{I} - \mathbf{K}_n \mathbf{H}) * E(w_n - \hat{w}_{n,n-1}) * E((\mathbf{K}_n \mathbf{v}_n)^T) = \\ &E(\mathbf{I} - \mathbf{K}_n \mathbf{H}) * 0 * E((\mathbf{K}_n \mathbf{v}_n)^T) = 0 \end{aligned} \right.$$

$$3 \left\{ \begin{aligned} &E(\mathbf{K}_n \mathbf{v}_n (w_n - \hat{w}_{n,n-1})^T (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T) = \\ &E(\mathbf{K}_n \mathbf{v}_n) * E((w_n - \hat{w}_{n,n-1})^T) * E((\mathbf{I} - \mathbf{K}_n \mathbf{H})^T) = \\ &E(\mathbf{K}_n \mathbf{v}_n) * 0 * E((\mathbf{I} - \mathbf{K}_n \mathbf{H})^T) = 0 \end{aligned} \right.$$

$$4 \left\{ E(\mathbf{v}_n \mathbf{v}_n^T) = \mathbf{R}_n \right.$$

A matriz \mathbf{R}_n representa as covariâncias das medições, ou seja, incerteza nas medições, enquanto que \mathbf{v}_n representa o erro da medição na iteração n . Portanto, $E(\mathbf{v}_n \mathbf{v}_n^T) = \mathbf{R}_n$ implica que a matriz de covariância das medições é igual à média ponderada das matrizes de covariância dos erros da medição. Logo, o cálculo de $\mathbf{P}_{n,n}$ se reduz a Equação 2.18.

$$\mathbf{P}_{n,n} = (\mathbf{I} - \mathbf{K}_n \mathbf{H}) \mathbf{P}_{n,n-1} (\mathbf{I} - \mathbf{K}_n \mathbf{H})^T + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T \quad (2.18)$$

Por fim, o cálculo do Ganho de Kalman é obtido pela minimização dos valores da diagonal principal da matriz de covariância $\mathbf{P}_{n,n}$ para que se tenha o valor mínimo da

variância estimada. Pela aplicação da Equação 2.18, juntamente com a propriedade de matriz transposta ($(\mathbf{AB})^T = \mathbf{B}^T \mathbf{A}^T$) obtemos

$$\begin{aligned} \mathbf{P}_{n,n} &= (\mathbf{P}_{n,n-1} - \mathbf{K}_n \mathbf{H} \mathbf{P}_{n,n-1}) (\mathbf{I} - \mathbf{H}^T \mathbf{K}_n^T) + \mathbf{K}_n \mathbf{R}_n \mathbf{K}_n^T = \\ &\mathbf{P}_{n,n-1} - \mathbf{P}_{n,n-1} \mathbf{H}^T \mathbf{K}_n^T - \mathbf{K}_n \mathbf{H} \mathbf{P}_{n,n-1} + \mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n) \mathbf{K}_n^T \end{aligned} \quad (2.19)$$

A soma da diagonal principal de uma matriz quadrática é chamada de traço da matriz e denotada por $tr(\mathbf{P}_{n,n})$. O objetivo é minimizar $tr(\mathbf{P}_{n,n})$ através das condições necessárias que produzam esse mínimo. O traço é então aplicado na Equação 2.19, derivado em relação a \mathbf{K}_n e definido como zero para minimizar o traço.

$$\begin{aligned} tr(\mathbf{P}_{n,n}) &= \\ tr(\mathbf{P}_{n,n-1}) - tr(\mathbf{P}_{n,n-1} \mathbf{H}^T \mathbf{K}_n^T) - tr(\mathbf{K}_n \mathbf{H} \mathbf{P}_{n,n-1}) + tr(\mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n) \mathbf{K}_n^T) \\ &= tr(\mathbf{P}_{n,n-1}) - 2tr(\mathbf{K}_n \mathbf{H} \mathbf{P}_{n,n-1}) + tr(\mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n) \mathbf{K}_n^T) \end{aligned}$$

A derivação da equação resultante, usando $\frac{d}{dA}(tr(AB)) = B^T$ e $\frac{d}{dA}(tr(ABA^T)) = 2AB$, simplifica novamente o cálculo [20].

$$\begin{aligned} \frac{d(tr(\mathbf{P}_{n,n}))}{d\mathbf{K}_n} &= \frac{d(tr(\mathbf{P}_{n,n-1}))}{d\mathbf{K}_n} - \frac{d(2tr(\mathbf{K}_n \mathbf{H} \mathbf{P}_{n,n-1}))}{d\mathbf{K}_n} + \\ \frac{d(tr(\mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n) \mathbf{K}_n^T))}{d\mathbf{K}_n} &= 0 - 2(\mathbf{H} \mathbf{P}_{n,n-1})^T + \\ 2\mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n) &= 0 \end{aligned}$$

Logo, temos que $(\mathbf{H} \mathbf{P}_{n,n-1})^T = \mathbf{K}_n (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n)$ e o ganho de Kalman é então dado pela Equação 2.20.

$$\begin{aligned} \mathbf{K}_n &= (\mathbf{H} \mathbf{P}_{n,n-1})^T (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n)^{-1} \\ &= \mathbf{P}_{n,n-1}^T \mathbf{H}^T (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n)^{-1} \\ &= \mathbf{P}_{n,n-1} \mathbf{H}^T (\mathbf{H} \mathbf{P}_{n,n-1} \mathbf{H}^T + \mathbf{R}_n)^{-1} \end{aligned} \quad (2.20)$$

2.3.2 Aplicação prática do filtro de Kalman multivariado

Esta seção apresenta um exemplo prático para ilustrar a aplicação do filtro de Kalman multivariado. O cenário descreve um robô autônomo se movendo num plano 2D e o objetivo é estimar sua posição atual com medições realizadas de 1 em 1 segundo. As iterações do processo são realizadas a cada medição, usando um mapeamento com o sensor LIDAR. O robô se move de acordo com a função $2^{\frac{x}{30}}$, variando o valor de x e mantendo a coordenada Y praticamente constante ($y \approx 0$). Em seguida, o robô faz uma curva para esquerda, aumentando os valores de x e y . Após a curva, o valor de y aumenta, mantendo o valor de x praticamente constante. O desvio padrão da aceleração é de $\alpha_a = 0.1m/s^2$ e o desvio padrão do erro de medição é de $\alpha_x = \alpha_y = 3m$.

Iteração 0

Inicialização:

No início do processo de medição não se sabe a localização atual do objeto. Por isso, os parâmetros de posição, velocidade e aceleração assumem valor 0, como apresentado no vetor de inicialização $\hat{w}_{0,0}$.

$$\hat{w}_{0,0} = \begin{bmatrix} x \\ \dot{x} \\ \ddot{x} \\ y \\ \dot{y} \\ \ddot{y} \end{bmatrix} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

Como o palpite inicial é uma estimativa sem embasamento, a incerteza da estimativa recebe um valor alto no cálculo do ganho de Kalman $P_{0,0}$ da primeira medição, como descrito na Subseção 2.3.1 no cálculo da extrapolação de covariância (Equação 2.14).

$$P_{0,0} = \begin{bmatrix} 500 & 0 & 0 & 0 & 0 & 0 \\ 0 & 500 & 0 & 0 & 0 & 0 \\ 0 & 0 & 500 & 0 & 0 & 0 \\ 0 & 0 & 0 & 500 & 0 & 0 \\ 0 & 0 & 0 & 0 & 500 & 0 \\ 0 & 0 & 0 & 0 & 0 & 500 \end{bmatrix}$$

A matriz de transição de estado F é dada a seguir, seguindo o intervalo de 1 segundo a cada iteração.

$$F = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \Delta 0.5\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} = \begin{bmatrix} 1 & 1 & 0,5 & 0 & 0 & 0 \\ 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 1 & 0,5 \\ 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

Em seguida, a matriz de processo de ruído Q é obtida de acordo com o desvio padrão de aceleração de $0.2m/s^2$. Como o nosso modelo não inclui dados de controle, que são correspondentes ao movimento do objeto de estudo, ou seja, comandos sobre o robô tais como virar a direita ou seguir reto, é necessário projetar a variância da aceleração α_x^2 usando a matriz de transição de estado [20, 15]. Logo, o cálculo é obtido pela fórmula

$Q = F * Q_{\hat{x}} * F^T$, onde as matrizes F e $Q_{\hat{x}}$ são dadas como segue.

$$Q_{\hat{x}} = \begin{bmatrix} 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix} * \alpha_{\hat{x}}^2 \quad F = \begin{bmatrix} 1 & \Delta t & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

A matriz resultante Q é então obtida como descrito a seguir.

$$Q = \begin{bmatrix} \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta t^4}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ 0 & 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} * \alpha_{\hat{x}}^2 = \begin{bmatrix} 0,0025 & 0,005 & 0,005 & 0 & 0 & 0 \\ 0,005 & 0,01 & 0,01 & 0 & 0 & 0 \\ 0,005 & 0,01 & 0,01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0,0025 & 0,005 & 0,005 \\ 0 & 0 & 0 & 0,005 & 0,01 & 0,01 \\ 0 & 0 & 0 & 0,005 & 0,01 & 0,01 \end{bmatrix}$$

Já a matriz de covariância de medição R é obtida com base no desvio padrão do erro de medição no valor de 3 metros.

$$R = \begin{bmatrix} \alpha_x^2 & 0 \\ 0 & \alpha_y^2 \end{bmatrix} = \begin{bmatrix} 9 & 0 \\ 0 & 9 \end{bmatrix}$$

Predição:

Com as matrizes inicializadas, o filtro de Kalman bidimensional realiza as predições, calculando as matrizes $\hat{w}_{1,0}$ e $P_{1,0}$, que são, respectivamente, as matrizes de extra-polação de estado e covariância.

$$\vec{w}_{1,0} = \vec{F} * \vec{w}_{0,0} = \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix}$$

$$\vec{P}_{1,0} = \vec{F} * \vec{P}_{0,0} * \vec{F}^T + \vec{Q} = \begin{bmatrix} 1125,0025 & 750,005 & 250,005 & 0 & 0 & 0 \\ 750,005 & 1000,01 & 500,01 & 0 & 0 & 0 \\ 250,005 & 500,01 & 500,01 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1125,0025 & 750,005 & 250,005 \\ 0 & 0 & 0 & 750,005 & 1000,01 & 500,01 \\ 0 & 0 & 0 & 250,005 & 500,01 & 500,01 \end{bmatrix}$$

Iteração 1

Medição:

Com a predição inicial calculada, a próxima etapa do processo mede a posição atual do robô no plano obtida pelo LIDAR.

$$\vec{z}_1 = \begin{bmatrix} 19,5 \\ 2,08 \end{bmatrix}$$

Estimativa:

Na sequência as estimativas dessa iteração são calculadas: o ganho de Kalman \vec{K}_1 , a estimativa do estado atual $\vec{w}_{1,1}$ e a covariância atual $\vec{P}_{1,1}$. A matriz \vec{H} , usada neste exemplo, é definida a seguir, com 6 dimensões (posição, velocidade e aceleração para x e y , respectivamente), onde apenas duas são dimensões mensuráveis (posição de x e y).

$$\vec{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

$$\vec{K}_1 = \vec{P}_{1,0} * \vec{H}^T * (\vec{H} * \vec{P}_{1,0} * \vec{H}^T + \vec{R})^{-1} = \begin{bmatrix} 0,9921 & 0 \\ 0,6614 & 0 \\ 0,2205 & 0 \\ 0 & 0,9921 \\ 0 & 0,6614 \\ 0 & 0,2205 \end{bmatrix}$$

O ganho de Kalman para a posição do robô, tanto para a coordenada x quanto para y , é de 0,9921. O valor alto no ganho de Kalman indica que, ou seja, quanto maior o ganho de Kalman mais o resultado da estimativa vai depender do valor medido em detrimento ao valor predito proporcionalmente. O valor resultante é de acordo com 99,21% do valor medido e 0,79% do valor predito.

Em seguida a estimativa do estado atual $w_{1,1}$ e a estimativa da covariância atual $P_{1,1}$ são calculadas.

$$\vec{w}_{1,1} = \vec{w}_{1,0} + \vec{K}_1 * (\vec{z}_1 - \vec{H} * \vec{w}_{1,0}) = \begin{bmatrix} 19,35 \\ 12,9 \\ 4,3 \\ 2,06 \\ 1,38 \\ 0,46 \end{bmatrix}$$

$$\vec{P}_{1,1} = (\vec{I} - \vec{K}_1 \vec{H}) \vec{P}_{1,0} (\vec{I} - \vec{K}_1 \vec{H})^T + \vec{K}_1 \vec{R} \vec{K}_1^T = \begin{bmatrix} 8,93 & 5,95 & 1,98 & 0 & 0 & 0 \\ 5,95 & 503,97 & 334,66 & 0 & 0 & 0 \\ 1,98 & 334,66 & 444,89 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8,93 & 5,95 & 1,98 \\ 0 & 0 & 0 & 5,95 & 503,97 & 334,66 \\ 0 & 0 & 0 & 1,98 & 334,66 & 444,89 \end{bmatrix}$$

Os valores da estimativa do estado atual do objeto, no vetor $w_{1,1}$, representam a posição, velocidade e aceleração do robô nas coordenadas x e y . Esses valores mostram de que forma o robô está se movimentando de uma iteração para outra. Já os valores para a estimativa da covariância na matriz $\vec{P}_{1,1}$, representam a incerteza associada ao estado estimado nesta iteração.

Dessa forma, representam a faixa de variação de cada valor, neste caso os valores da posição de x e y podem variar do valor resultado de $\vec{w}_{1,1}$ por 8,93 positiva ou negativamente, já o valor da aceleração de ambos pode variar 444,89 positiva ou negativamente. É possível perceber que a matriz $\vec{P}_{n,n}$, conforme as iterações são realizadas, tende a diminuir seus valores num geral, pois a cada iteração do filtro o seu nível de incerteza tende a diminuir.

Predição:

Para finalizar a primeira iteração é necessário realizar a predição do próximo estado, que será usada para estimar da próxima iteração, juntamente com a leitura da posição real do robô.

$$\vec{w}_{2,1} = \vec{F} * \vec{w}_{1,1} = \begin{bmatrix} 34,39 \\ 17,2 \\ 4,3 \\ 3,67 \\ 1,83 \\ 0,46 \end{bmatrix}$$

$$\vec{P}_{2,1} = \vec{F} * \vec{P}_{1,1} * \vec{F}^T + \vec{Q} = \begin{bmatrix} 972,68 & 1236,35 & 559,1 & 0 & 0 & 0 \\ 1236,35 & 1618,2 & 779,57 & 0 & 0 & 0 \\ 559,1 & 779,57 & 444,9 & 0 & 0 & 0 \\ 0 & 0 & 0 & 972,68 & 1236,36 & 559,1 \\ 0 & 0 & 0 & 1236,35 & 1618,2 & 779,57 \\ 0 & 0 & 0 & 559,1 & 779,57 & 444,9 \end{bmatrix}$$

Iteração 2

Medição:

$$\vec{z}_2 = \begin{bmatrix} 25,24 \\ 1,92 \end{bmatrix}$$

Estimativa:

$$\vec{K}_2 = \vec{P}_{2,1} * \vec{H}^T * (\vec{H} * \vec{P}_{2,1} * \vec{H}^T + \vec{R})^{-1} = \begin{bmatrix} 0,9908 & 0 \\ 1,2594 & 0 \\ 0,5695 & 0 \\ 0 & 0,9908 \\ 0 & 1,2594 \\ 0 & 0,5695 \end{bmatrix}$$

$$\vec{w}_{2,2} = \vec{w}_{2,1} + \vec{K}_2 * (\vec{z}_2 - \vec{H} * \vec{w}_{2,1}) = \begin{bmatrix} 25,32 \\ 5,67 \\ -0,91 \\ 1,94 \\ -0,37 \\ -0,54 \end{bmatrix}$$

$$\vec{P}_{2,2} = (\vec{I} - \vec{K}_2 \vec{H}) \vec{P}_{2,1} (\vec{I} - \vec{K}_2 \vec{H})^T + \vec{K}_2 \vec{R} \vec{K}_2^T = \begin{bmatrix} 8,92 & 11,33 & 5,13 & 0 & 0 & 0 \\ 11,33 & 61,1 & 75,42 & 0 & 0 & 0 \\ 5,13 & 75,42 & 126,48 & 0 & 0 & 0 \\ 0 & 0 & 0 & 8,92 & 11,33 & 5,13 \\ 0 & 0 & 0 & 11,33 & 61,1 & 75,42 \\ 0 & 0 & 0 & 5,13 & 75,42 & 126,48 \end{bmatrix}$$

Percebe-se que a incerteza associada a cada medição e suas covariâncias diminuiu da matriz $\vec{P}_{1,1}$ para a $\vec{P}_{2,2}$, significando que os valores estimados estão convergindo para valores mais precisos/próximos dos reais.

Predição:

$$\vec{w}_{3,2} = \vec{F} * \vec{w}_{2,2} = \begin{bmatrix} 30,54 \\ 4,76 \\ -0,91 \\ 1,3 \\ -0,9 \\ -0,54 \end{bmatrix}$$

$$\vec{P}_{3,2} = \vec{F} * \vec{P}_{2,2} * \vec{F}^T + \vec{Q} = \begin{bmatrix} 204,86 & 253,94 & 143,79 & 0 & 0 & 0 \\ 253,94 & 338,43 & 201,91 & 0 & 0 & 0 \\ 143,79 & 201,91 & 126,49 & 0 & 0 & 0 \\ 0,01 & 0,01 & 0 & 204,86 & 253,94 & 143,79 \\ 0,01 & 0,01 & 0 & 253,94 & 338,43 & 201,91 \\ 0 & 0 & 0 & 143,79 & 201,91 & 126,49 \end{bmatrix}$$

Iteração 35

Passando alguns instantes, realizando iteração por iteração, é chegada na operação atual, é possível perceber que, principalmente nas matrizes do ganho de Kalman \vec{K}_{35} e da estimativa da covariância $\vec{P}_{35,35}$, os valores estimados e preditos do filtro acabaram convergindo para valores mais condizentes do estado real do robô.

Medição:

$$\vec{z}_{35} = \begin{bmatrix} 189,02 \\ 80,72 \end{bmatrix}$$

Estimativa:

$$\vec{K}_{35} = \vec{P}_{35,34} * \vec{H}^T * (\vec{H} * \vec{P}_{35,34} * \vec{H}^T + \vec{R})^{-1} = \begin{bmatrix} 0,4747 & 0 \\ 0,1515 & 0 \\ 0,0242 & 0 \\ 0 & 0,4747 \\ 0 & 0,1515 \\ 0 & 0,0242 \end{bmatrix}$$

$$\vec{w}_{35,35} = \vec{w}_{35,34} + \vec{K}_{35} * (\vec{z}_{35} - \vec{H} * \vec{w}_{35,34}) = \begin{bmatrix} 189,55 \\ 4,76 \\ -0,04 \\ 79,19 \\ 8,07 \\ 0,56 \end{bmatrix}$$

$$\vec{P}_{35,35} = (\vec{I} - \vec{K}_{35} \vec{H}) \vec{P}_{35,34} (\vec{I} - \vec{K}_{35} \vec{H})^T + \vec{K}_{35} \vec{R} \vec{K}_{35}^T = \begin{bmatrix} 4,27 & 1,36 & 0,22 & 0 & 0 & 0 \\ 1,36 & 0,74 & 0,17 & 0 & 0 & 0 \\ 0,22 & 0,17 & 0,05 & 0 & 0 & 0 \\ 0 & 0 & 0 & 4,27 & 1,37 & 0,22 \\ 0 & 0 & 0 & 1,36 & 0,74 & 0,17 \\ 0 & 0 & 0 & 0,22 & 0,16 & 0,05 \end{bmatrix}$$

Predição:

$$\vec{\hat{w}}_{36,35} = \vec{F} * \vec{\hat{w}}_{35,35} = \begin{bmatrix} 194,29 \\ 4,72 \\ -0,04 \\ 87,55 \\ 8,64 \\ 0,56 \end{bmatrix}$$

$$\vec{P}_{36,35} = \vec{F} * \vec{P}_{35,35} * \vec{F}^T + \vec{Q} = \begin{bmatrix} 8,13 & 2,59 & 0,410 & 0 & 0 & \\ 2,59 & 1,13 & 0,230 & 0 & 0 & \\ 0,41 & 0,23 & 0,060 & 0 & 0 & \\ 0 & 0 & 0 & 8,14 & 2,6 & 0,42 \\ 0 & 0 & 0 & 2,59 & 1,13 & 0,23 \\ 0 & 0 & 0 & 0,41 & 0,23 & 0,06 \end{bmatrix}$$

Resultados

A Figura 6 apresenta o gráfico com os resultados da aplicação do filtro de Kalman bidimensional no cenário descrito. A linha *Real* mostra a a real posição do robô no ambiente. A linha *Medido* é obtida pelos valores mensurados durante as iterações. A linha *Estimado* mostra os valores estimados pelo filtro ao longo das iterações Por fim, a linha do *Intervalo de Confiança* remete a faixa de valores confiáveis do filtro. De acordo com a incerteza associada $\vec{P}_{n,n}$ (quanto mais alta a incerteza maior o intervalo de confiança) é gerada uma faixa em torno dos valores *Reais* que representa 95% de precisão, ou seja, valores dentro dessa faixa são os 5% mais precisos.

Percebe-se que todos os valores estimados permaneceram dentro dessa faixa, portanto a execução do filtro de Kalman pode ser considerada confiável.

O gráfico também mostra que o resultado obtido pelo SLAM multivariado para este exemplo tende a ser mais próximo do valor real quando comparado com os valores medidos pelo equipamento que do aqueles obtidos pelo exemplo do filtro de Kalman unidimensional apresentado na Figura 5. Logo, as estimativas do modelo bidimensional são mais precisas do que os sensores que capturam as informações sobre o objeto.

O motivo pelo qual neste exemplo está mais elucidado e visual a maior precisão do valor estimado em relação ao valor medido se dá pelo fato de que o modelo bidimensional apresenta duas medidas distintas, dessa forma, uma medição que erre 1% em relação aos valores reais é mais visível do que a mesma medição em apenas uma dimensão, ou seja, no caso do bidimensional as imprecisões são tanto em x quanto em y , permitindo uma melhor visualização do funcionamento do filtro.

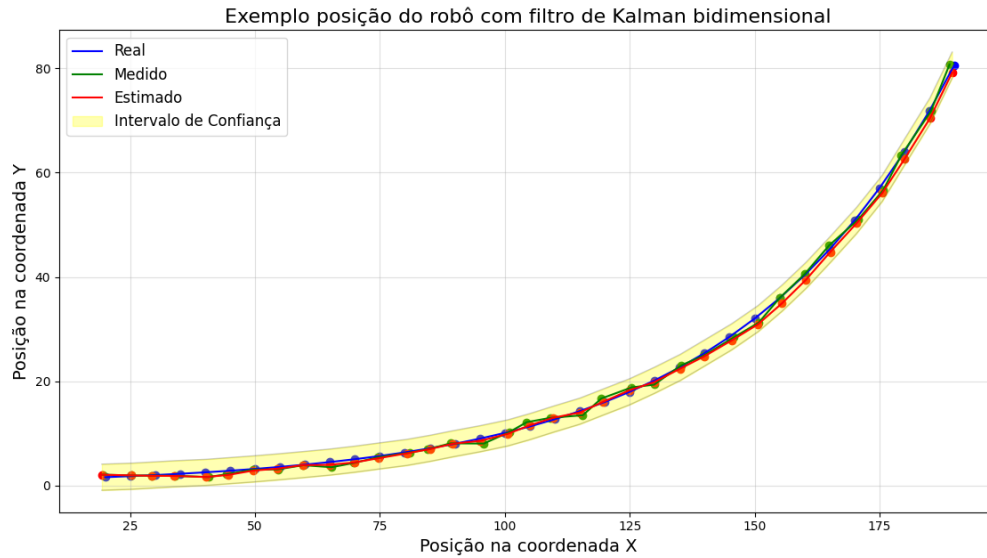


Figura 6 – Resultados do exemplo prático com filtro de Kalman bidimensional

2.3.3 Projetando o Ambiente Tridimensional para Bidimensional

A predição em sistemas tridimensionais muitas vezes ocorre com a transformação de um ambiente 3D para o espaço 2D, para então ser aplicado o modelo bidimensional. Essa projeção do plano, também chamada de planificação, é uma etapa crucial em muitas aplicações que envolvem ambientes reais. O objetivo principal dessa planificação é a representação de informações 3D num formato 2D, anulando, ou ignorando, o eixo z , de modo a evitar que medições e predições sobre este plano sejam calculadas durante as etapas do processo de execução do filtro. Assim, este processo de planificação quando aplicado nas múltiplas dimensões que estão relacionadas nos cálculos realizados, como por exemplo na determinação do valor da covariância, impede que erros acumulados e aleatórios calculados de medições sobre o plano z sejam considerados nas etapas do processo de predição.

Esse processo pode ou não ser possível de se realizar dependendo do objetivo da aplicação. No caso de simulação de robôs em logística industrial ou uso doméstico, como os aspiradores de pó automático, que se locomovem apenas em solo, a terceira dimensão (altura) nas equações do filtro seriam desnecessárias e, possivelmente, prejudiciais no cálculos. Já em aplicações com drones e robôs submersíveis, essa estratégia de eliminar o plano z não pode ser aplicada. Além disso, o método de planificação mencionado, e utilizado no decorrer deste trabalho, tem relação com uma projeção ortogonal que gera uma visão panorâmica superior do ambiente (uma “vista de cima”). Todos os objetos presentes no ambiente medido se tornam planificados sobre o eixo z , como por exemplo uma parede, que nesta visão se torna um contorno sem a medida de altura, mantendo apenas a largura e o comprimento.

A Figura 7 ilustra essa transformação na simulação de um ambiente real. O sensor LIDAR percorre apenas em duas dimensões, largura e comprimento, ignorando a altura. As medições dos objetos no ambiente são enviadas ao filtro diretamente planejado, com apenas duas dimensões. Como se observa na figura, o sensor LIDAR 2D realiza a leitura, localizando as paredes e obstáculos, num plano 2D para obter o mapa gerado e representado na lupa no canto inferior direito.

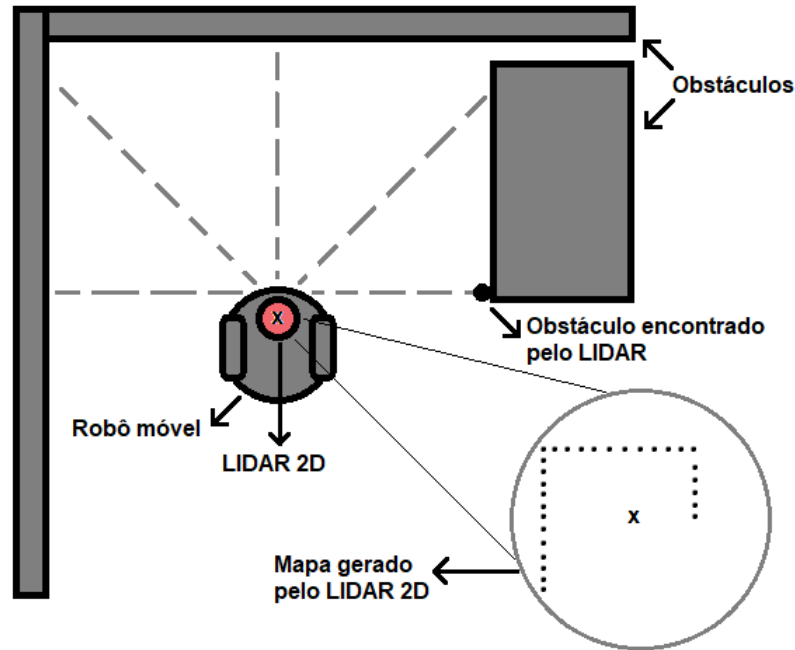


Figura 7 – Leitura planejada de um LIDAR 2D em um ambiente 3D.

Portanto, uma solução num ambiente tridimensional pode ser obtida através das equações apresentadas na Subseção 2.3.1 usando o modelo do filtro de Kalman bidimensional.

2.4 Algoritmos e estrutura de código

Esta seção apresenta as estruturas de código do filtro de Kalman bivariado num contexto multivariado para o problema do SLAM. Primeiro é apresentada as inicializações das matrizes, definindo seus respectivos valores de acordo com o que será utilizado no filtro, em sequência é apresentado o primeiro algoritmo, ele que é relativo às leituras realizadas pelo LIDAR, retornando um vetor de pontos que são as leituras dos objetos encontrado por ele, e, por fim, é apresentado o terceiro algoritmo, que é a realização das iterações do filtro em si.

A inicialização do filtro define as matrizes \vec{F} , \vec{Q} e \vec{R}_n , usadas ao longo do processo, com valores constantes [20, 29, 15], bem como as matrizes que representam o palpite inicial sobre a posição inicial do objeto, $\vec{w}_{0,0}$, e da covariância estimada, $\vec{P}_{0,0}$.

Veja o Algoritmo ??.

Para as variâncias estimadas, que são definidas na matriz $\vec{P}_{0,0}$, é comum sempre iniciá-las com valores altos pois, por mais próximo que esteja o palpite inicial sobre posição, velocidade e aceleração, o filtro de Kalman acaba convergindo sempre para um valor próximo, não importando quão distante estão os valores do palpite inicial.

Começando com as inicializações das matrizes, os dados referentes para suas definições são: período de medições Δt , desvio padrão da aceleração aleatória σ_a , desvio padrão do erro de medição σ_{xm} e σ_{ym} , palpite inicial de posição, velocidade e aceleração dos eixos x e y do objeto de estudo $\vec{w}_{n,n}$ e a variância estimada da posição dos eixos x e y do objeto de estudo p_x e p_y . Com isso, as matrizes ficam de acordo como segue:

$$\vec{w}_n = \begin{bmatrix} x_n \\ y_n \\ \dot{x}_n \\ \dot{y}_n \\ \ddot{x}_n \\ \ddot{y}_n \end{bmatrix}$$

$$\vec{P}_{n,n} = \begin{bmatrix} p_x & 0 & 0 & 0 & 0 & 0 \\ 0 & p_x & 0 & 0 & 0 & 0 \\ 0 & 0 & p_{\ddot{x}} & 0 & 0 & 0 \\ 0 & 0 & 0 & p_y & 0 & 0 \\ 0 & 0 & 0 & 0 & p_{\dot{y}} & 0 \\ 0 & 0 & 0 & 0 & 0 & p_{\ddot{y}} \end{bmatrix}$$

$$\vec{F} = \begin{bmatrix} 1 & \Delta t & 0.5\Delta t^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \Delta 0.5\Delta t^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & 1 \end{bmatrix}$$

$$\vec{Q} = \begin{bmatrix} \frac{\Delta t^2}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} & 0 & 0 & 0 \\ \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t & 0 & 0 & 0 \\ \frac{\Delta t^2}{2} & \Delta t & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{\Delta t^2}{4} & \frac{\Delta t^3}{2} & \frac{\Delta t^2}{2} \\ 0 & 0 & 0 & \frac{\Delta t^3}{2} & \Delta t^2 & \Delta t \\ 0 & 0 & 0 & \frac{\Delta t^2}{2} & \Delta t & 1 \end{bmatrix} * \sigma_x^2$$

$$\vec{R}_n = \begin{bmatrix} \sigma_{xm}^2 & 0 \\ 0 & \sigma_{ym}^2 \end{bmatrix}$$

$$\vec{H} = \begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}$$

O Algoritmo 1 é responsável pelo processamento dos dados provenientes do sensor LIDAR [19, 30]. Como o LIDAR, a cada iteração, de um em um segundo juntamente com as iterações do filtro, realiza uma volta completa (360 graus) sobre ele mesmo, é necessário dividir por quantos graus existem de diferença entre uma leitura e outra, esse valor é dado de acordo com a especificação de cada sensor LIDAR, por exemplo se um LIDAR mede os pontos girando em torno de si mesmo com uma diferença de um grau por ponto, são totalizadas 360 leituras, se a diferença é de meio grau por ponto, são totalizadas 720 leituras e assim por diante, onde o *for* percorre de 0 até $\frac{360}{\theta}$ pois, sabendo que o LIDAR, a cada iteração, realiza uma volta completa (360 graus) sobre ele mesmo, é necessário dividir por quantos graus existem de diferença entre uma leitura e outra. Por exemplo, suponha que o LIDAR esteja configurado para uma diferença de 1 grau entre as leituras, O número de pontos lidos retornados pelo LIDAR é de $\frac{360}{1} = 360$ pontos. Num outro cenário, se a diferença é de 0,5 graus entre as leituras, o número de pontos retornados é de $\frac{360}{0,5} = 720$ pontos. Dessa forma, o Algoritmo 1 calcula as coordenadas x, y sobre as leituras do LIDAR e armazena no vetor $vet[(x_1, y_1); (x_2, y_2); \dots; (x_i, y_i)]$. A função `getLidarDist` dentro do algoritmo 1 recebe como parâmetro o ângulo em radiano e retorna a distância obtida pelo LIDAR neste ângulo e as funções `cos` e `sin` retornam, respectivamente, o valor do cosseno e seno do ângulo em radiano passado.

Algorithm 1: LIDAR_Data_3D-2D(θ)

```

1 Function Lidar( $\theta$ ):
2   vet[];
3   for  $i \in 0 \dots \frac{360}{\theta}$  do
4     rad =  $i * \theta * \frac{\pi}{180}$ ;
5     distance = getLidarDist(rad);
6     x = distance * cos(rad);
7     y = distance * sin(rad);
8     vet[i] = (x,y);

```

Já o Algoritmo 2 descreve a computação das estimativas e previsões do filtro [28], com base nos dados obtidos pela inicialização das matrizes e do Algoritmos1, ele retorna como saída a estimativa do estado atual $\vec{w}_{n,n}$ e a estimativa da covariância atual $\vec{P}_{n,n}$. Neste algoritmo são calculados os valores de posição e covariância, estimados e preditos, estes apenas que são armazenados entre iterações, e a matriz de ganho de Kalman, além de receber os dados de medição provenientes do LIDAR.

Algorithm 2: SLAM_Kalman_3D-2D($\Delta t, \sigma_{xm}, \sigma_{ym}, \vec{w}_{n,n}, \vec{H}, p_x, p_y, \theta$)

```

1 n = 0;
2 while verdadeiro do
3    $\vec{w}_{n+1,n} = \vec{F} * \vec{w}_{n,n}$  ; // Predição da posição
4    $\vec{P}_{n+1,n} = \vec{F} * \vec{P}_{n,n} * \vec{F}^T + \vec{Q}$  ; // Predição da covariância estimada
5   n = n + 1 ; // Próxima iteração
6    $\vec{z}_n = \text{LIDAR\_Data\_3D-2D}(\theta)$  ; // Medição da posição feita pelo
   sensor
7    $\vec{K}_n = \vec{P}_{n,n-1} * \vec{H}^T * (\vec{H} * \vec{P}_{n,n-1} * \vec{H}^T + \vec{R}_n)^{-1}$  ; // Estimativa do ganho
   de Kalman
8    $\vec{w}_{n,n} = \vec{w}_{n,n-1} + \vec{K}_n * (z_n - H * \vec{w}_{n,n-1})$  ; // Estimativa da posição
9    $\vec{P}_{n,n} = (I - K_n * H) * \vec{P}_{n,n-1} * (I - K_n * H)^T + K_n * R_n * K_n^T$  ;
   // Estimativa da covariância

```

O algoritmo 2 apresentado tem um custo computacional geral, no pior caso, de acordo com a função de complexidade de tempo $O(n^3)$, nas multiplicações de matrizes, e de acordo com a função de complexidade de espaço $O(n^2)$, ambos para com matrizes de tamanho $n \times n$ e $n \times m$, sendo assim um algoritmo eficiente [31], tendo em vista que as operações mais pesadas são as de adição, subtração e multiplicação de matrizes em questão de tempo e, conseqüentemente, o armazenamento de todas elas, em questão de espaço. O tamanho de todas as matrizes é de ordem 6×6 , devido aos parâmetros de posição, velocidade e aceleração conforme apresentado, com exceção das matrizes de predição e estimativa de estado, $\vec{w}_{n+1,n}$ e $\vec{w}_{n,n}$, com tamanho 6×1 , e da matriz de covariância das medições, \vec{R}_n , com tamanho 2×2 .

REFERÊNCIAS

- [1] BRYSON, J.; WINFIELD, A. Standardizing ethical design for artificial intelligence and autonomous systems. *Computer, IEEE*, v. 50, n. 5, p. 116–119, 2017.
- [2] OTA, J. Multi-agent robot systems as distributed autonomous systems. *Advanced engineering informatics*, Elsevier, v. 20, n. 1, p. 59–70, 2006.
- [3] HA, Q.; YEN, L.; BALAGUER, C. Robotic autonomous systems for earthmoving in military applications. *Automation in Construction*, Elsevier, v. 107, p. 102934, 2019.
- [4] AULINAS, J. et al. The slam problem: a survey. *Artificial Intelligence Research and Development*, IOS Press, p. 363–371, 2008.
- [5] GRISSETTI, G. et al. A tutorial on graph-based slam. *IEEE Intelligent Transportation Systems Magazine*, IEEE, v. 2, n. 4, p. 31–43, 2010.
- [6] ROBINSON, E. A.; TREITEL, S. Principles of digital wiener filtering. *Geophysical Prospecting*, European Association of Geoscientists & Engineers, v. 15, n. 3, p. 311–332, 1967.
- [7] KALMAN, R. E. A new approach to linear filtering and prediction problems. *Journal of Basic Engineering*, ASME, v. 82, n. 1, p. 35, 1960. Disponível em: <<http://dx.doi.org/10.1115/1.3662552>>.
- [8] HUANG, S.; DISSANAYAKE, G. Convergence and consistency analysis for extended kalman filter based slam. *IEEE Transactions on robotics*, IEEE, v. 23, n. 5, p. 1036–1049, 2007.
- [9] HUANG, G. P.; MOURIKIS, A. I.; ROUMELIOTIS, S. I. Analysis and improvement of the consistency of extended kalman filter based slam. In: IEEE. *2008 IEEE International Conference on Robotics and Automation*. [S.l.], 2008. p. 473–479.
- [10] SMITH, R. C.; CHEESEMAN, P. On the representation and estimation of spatial uncertainty. *The international journal of Robotics Research*, Sage Publications Sage CA: Thousand Oaks, CA, v. 5, n. 4, p. 56–68, 1986.
- [11] DURRANT-WHYTE, H.; BAILEY, T. Simultaneous localization and mapping: part i. *IEEE robotics & automation magazine*, IEEE, v. 13, n. 2, p. 99–110, 2006.
- [12] CHATILA, R. Robot mapping: An introduction. *Robotics and cognitive approaches to spatial mapping*, Springer, p. 9–12, 2008.
- [13] MOSHAYEDI, A. J.; JINSONG, L.; LIAO, L. Agv (automated guided vehicle) robot: Mission and obstacles in design and performance. *Journal of Simulation and Analysis of Novel Technologies in Mechanical Engineering*, Islamic Azad University, Khomeinishahr Branch, v. 12, n. 4, p. 5–18, 2019.
- [14] SAYERS, C.; SAYERS, C. *Remote control robotics*. [S.l.]: Springer, 1999.
- [15] RIBEIRO, M. I. Kalman and extended kalman filters: Concept, derivation and properties. *Institute for Systems and Robotics*, v. 43, n. 46, p. 3736–3741, 2004.

- [16] THRUN, S. Particle filters in robotics. In: CITESEER. *UAI*. [S.l.], 2002. v. 2, p. 511–518.
- [17] SCHUSTER, F. et al. Landmark based radar slam using graph optimization. In: IEEE. *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC)*. [S.l.], 2016. p. 2559–2564.
- [18] ENGELHARD, N. et al. Real-time 3d visual slam with a hand-held rgb-d camera. In: *Proc. of the RGB-D Workshop on 3D Perception in Robotics at the European Robotics Forum, Vasteras, Sweden*. [S.l.: s.n.], 2011. v. 180, p. 1–15.
- [19] NAM, D. V.; GON-WOO, K. Solid-state lidar based-slam: A concise review and application. In: IEEE. *2021 IEEE International Conference on Big Data and Smart Computing (BigComp)*. [S.l.], 2021. p. 302–305.
- [20] BECKER, A. *KALMAN FILTER from the Ground Up*. 1. ed. [S.l.]: 2Checkout, 2023. ISBN 978-965-598-439-2.
- [21] LI, A. et al. From timing variations to performance degradation: Understanding and mitigating the impact of software execution timing in slam. In: IEEE. *2022 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2022. p. 13308–13315.
- [22] HSU, C.-P. et al. A review and perspective on optical phased array for automotive lidar. *IEEE Journal of Selected Topics in Quantum Electronics*, IEEE, v. 27, n. 1, p. 1–16, 2020.
- [23] ABANAY, A.; MASMOUDI, L.; ANSARI, M. E. A calibration method of 2d lidar-visual sensors embedded on an agricultural robot. *Optik*, Elsevier, v. 249, p. 168254, 2022.
- [24] FANG, H.-T.; HUANG, D.-S. Noise reduction in lidar signal based on discrete wavelet transform. *Optics Communications*, Elsevier, v. 233, n. 1-3, p. 67–76, 2004.
- [25] KANEKO, M. et al. Mask-slam: Robust feature-based monocular slam by masking using semantic segmentation. In: *Proceedings of the IEEE conference on computer vision and pattern recognition workshops*. [S.l.: s.n.], 2018. p. 258–266.
- [26] ALEXANDROV, S. *PointCloudLibrary pcl*. 1. ed. [S.l.]: github, 2013.
- [27] ESCOBAR, C. S. *Distribuciones de probabilidad*. RPubS, RPubS, p. 36, 2016.
- [28] GREWAL, M. S.; ANDREWS, A. P. Applications of kalman filtering in aerospace 1960 to the present [historical perspectives]. *IEEE Control Systems Magazine*, IEEE, v. 30, n. 3, p. 69–78, 2010.
- [29] CIVERA, J. et al. 1-point ransac for extended kalman filtering: Application to real-time structure from motion and visual odometry. *Journal of field robotics*, Wiley Online Library, v. 27, n. 5, p. 609–631, 2010.
- [30] YOKOZUKA, M. et al. Litamin: Lidar-based tracking and mapping by stabilized icp for geometry approximation with normal distributions. In: IEEE. *2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*. [S.l.], 2020. p. 5143–5150.

- [31] WELCH, G.; BISHOP, G. et al. An introduction to the kalman filter. Chapel Hill, NC, USA, 1995.