MPU - Raw data

Este script em matlab realiza a leitura de dados do MPU e permite a exibição dos dados brutos, e diversos processamentos de modo modular, tais como filtros e outros, podendo o usuário alterar os dados que deseja visualizar e diversos parâmetros do processamento.

O script tem por objetivo apresentar de forma simples os dados e cálculos envolvidos na leitura e fusão de dados do MPU, por isso possui bastante cálculos redundântes e não otimizados, sendo que muitos deles são implementação própria no lugar do uso de bibliotecas nativas, para deixar o mais claro possível os cálculos realizados.

Sumário

Features

- Layout modular
- Dados brutos de Aceleração, Giro/s nos 3 eixos
- Média movel dos dados brutos
- Posição angular relativa, utilizando dados de Giro/s (Posição não relativa aos eixos iniciais/da terra)
- Posição angular absoluta (tilt), utilizando dados de Giro/s, com uso de matriz de rotação (Posição relativa aos eixos iniciais)
- Posição angular absoluta (tilt), utilizando dados do Acelerômetro, com base em matriz de rotação
- Estimativa de posição angular (tilt), usando fusão de dados (Filtro de Kalman e filtro complementar)
- Estimativa de posição angular (tilt), utilizando filtro de madgwick
- Alteração de parâmetro dos filtros variável
- Estimativa de aceleração removendo vetor gravitacional (Utilizando estimativa de posição angular com base em filtro de kalman)
- Estimativa de Velocidade e posição no espaço, com base na aceleração sem vetor gravitacional
- Plotagem em tempo real (Com delay, sugeito a velocidade de processamento da maquina)
- Leitura de arquivos e porta serial

Features futuras

- Plotagem de quartenios
- Plotagem de representação do tilt em 3D
- Plotagem de representação do deslocamento em 3D
- Obtenção de variáveis do MPU (escalas, frequência de amostragem, bias) junto com a leitura dos dados
- Mudar o eixo temporal dos plots para ser medido em segundos e não em número de amostras

Como usar

Nas seções a seguir temos uma breve explicação do uso do script, caso prefira também há um vídeos explicativo: https://youtu.be/1yQjMueZ2hk

Formato de entrada dos dados

O formato de entrada dos dados é o mesmo para arquivo e porta serial, ele deve conter um demarcador de inicio (podendo conter qualquer dado antes), e um demarcador de final (podendo conter qualquer dado depois), entre o meio dos indicadores se encontram os dados, onde cada linha é uma amostra de todos os sensores separado por ; como segue abaixo

...
start
ax; ay; az; t; gx; gy; gz;
ax; ay; az; t; gx; gy; gz;
...
ax; ay; az; t; gx; gy; gz;
ax; ay; az; t; gx; gy; gz;
fim
...

onde, ax, ay e az são leitura do acelerômetro em X,Y e Z respectivamente gx, gy e gz são leitura do giroscopio em X,Y e Z respectivamente t é a temperatura

> Note que atualmente ainda não é levado em consideração os dados do magnetômetro

Leitura da porta serial

Para fazer leitura dos dados via porta serial, defina os parâmetros da fonte de leitura da seguinte forma:

- **read_from_serial**: true
- **serial_COM**: O número da porta COM que esta conectado o arduino. E.x.: 'COM5'
- **serial_baudrate**: O valor de taxa de transmissão configurado no arduino. E.x.: 115200

Leitura de um arquivo

Para fazer leitura dos dados via arquivo local, defina os parâmetros da 'fonte de leitura' da seguinte forma:

- **read_from_serial**: false
- **file_full_path**: Endereço do seu arquivo, relativo ao arquivo main.m. E.x.: 'Dados/My_file.txt'
- **file_simulated_freq**: Para o caso de simular uma plotagem em tempo real, usando arquivos defina este parâmetro (veja mais a respeito na seção de plotagem em tempo real)

Leitura e plotagem simultânea (tempo real)

Tanto a leitura de arquivo quando a leitura da porta serial podem realizar a plotâgem em tempo real, ou seja, ler os dados e plotar de forma intercalada, porém não é exatamente simultâneo, sendo executados em sequência de forma intervalada, para isso deve-se definir a frequência de plotagem e, levar em consideração que serão realizados calculos neste meio tempo, portanto, caso note algum erro ou garga-lo diminua a frequencia de plotagem ou mude para uma abordagem sem plotagem em tempo real.

Para definir uma abordagem de plotagem em tempo real defina as váriaveis de 'Plotagem' da seguinte maneira:

- **plot_in_real_time**: true
- **freq_render**: frequência em Hz de atualização dos plot. E.x.: 5, significa que a os plots serão atualizados a cada 200 milissegundos

Para o caso de leitura via arquivo você ainda pode definir o parâmetro 'file_simulated_freq' em 'Fonte de leitura' simulando o que seria a taxa de amostragem do MPU, entretanto essa funcionalidade não esta muito funcional, sendo bem mais lenta do que o real, portanto recomenda-se deixa-la sempre como Inf de Infinito.

Leitura e plotagem isoladamente

Assim como temos a abordagem de leitura em tempo real, podemos também realizar a abordagem tradicional, onde, reliza-se primeiro a leitura de todos os dados, até encontrar o demarcados de fim, depois é feito todos os calculos e, somente no fim é feito a plotagem, obviamente é uma abordagem muito mais rápida do que a plotagem em tempo real já que não tem o delay de plotar varias vezes por segundo.

Para relizar esta configuração basta definir o parâmetro em 'Plotagem', da seguinte forma:

- **plot_in_real_time**: false

Alteração de layout

Caso você queira alterar o layout, visualizar um gráfico em tamanho maior, remover ou adicionar gráficos, você pode faze-lo alterando o parâmetro 'layout' em 'Plotagem', você pode utilizar um ou mais dos items de layout disponíveis.

A configuração do parâmetro é bem simples, o parâmetro em si é uma matriz de tamanho variável, onde cada celula plota um item ou fica vazio. Para deixar um item maior que os demais, basta unir as celulas adjacentes (em vertical ou horizontal), repetindo o nome do item desejado, abaixo segue alguns exemplos:

Plotando 1 único item em tela cheia: layout= {... Acel... };

Plotando 2 linhas e 3 colunas, com Aceleração ocupado 4 das 6 celulas disponíveis e velocidade ocupando o espaço vertical: layout= {...

```
    Acel,       Acel,     Vel;...
    Acel,       Acel,     Vel;...
```

};

Caso deseje também é possível utilizar esta abordagem com multiplas janelas, entretanto requer um pouquinho de código, veja na seção de criação de janelas de plotagem para mais detalhes.

Por fim, também é possível ajustar a janela de amostras visiveis, sendo este tamanho de janela comum a todos os plots, basta definir o parâmetro 'max_size' em 'Amostragem', este dado é medido em número de amostras.

Items disponíveis no layout

- **Vazio** - Deixa a celula vazia
- **Acel** - Aceleração x,y,z
- **Vel** - Velocidade x,y,z
- **Space** - Espaço percorrido x,y,z
- **Gvel** - Velocidade angular em x,y,z
- **Gdeg** - Posição angular em x,y,z relativo (em relação a posição anterior)
- **Gtilt** - Posição angular em x,y,z absoluto (em relação aos eixos iniciais)
- **Mag** - Magnetometro
- **FusionTilt** - Posição angular em x,y,z absoluto, usando aceleração e magnetometro
- **CompTilt** - Posição angular usando o filtro complementar
- **KalmanTilt** - Posição angular usando o filtro de kalman
- **MadgwickTilt** - Posição angular usando o filtro de madgwick
- **Space3D** - Posição em um espaço 3D
- **Tilt3D** - Posição angular atual usando um objeto 3D
- **Acel_G** - Aceleração desconsiderando a gravidade (utilizando o melhor filtro p/ remove-la, - pois é necessário saber bem a posição angular p/ isso)

Alteração dos bias, escalas e frequência de amostragem

Atualmente o programa não realiza a leitura do bias diretamente do arquivo/arduino, cabendo ao usuário se desejado, adicionar o valor de bias para correção deste erro nos sensores, nos parâmetros 'Constantes do sensor' é possível encontrar

variáveis para definir isto bem como definir as escaladas utilizadas pelo MPU.

Por fim o MPU é configurado com uma frequência de amostragem, o parâmetro 'freq_sample' em 'Amostragem' que corresponde a este dado, e deve ser alterado para corresponder a frequência de amostragem do dado, sendo ele de fundamental importância nos calculos realizados.

> NOTE: a frequência de amostragem deve corresponder a frequência configurado no MPU para amostrar os dados em questão, tanto para leitura serial quanto de arquivo, assim como as escalas e bias.

Ajustes de média movel

A média movel é realizada nos dados brutos obtidos pelos sensores, com o objetivo de amortizar os erros de leitura, este dado pode ser mudado alterando o parâmetro 'window_k' em 'Media movel parametros'

Ajustes de filtro complementar

O filtro complementar é responsável por fundir dados do acelerômetro, magnetômetro e giroscópio, para uma melhor estimativa da posição angular do corpo (tilt).

O parâmetro 'mu' presente em 'Variável de ajuste do filtro complementar' controla quão rápido este dado irá convergir para os dados de magnetômetor e acelerômetro.

> NOTE: quanto menor o valor de 'mu' mais lento a convergência entretanto, menor o ruido causado pelo magnetômetor e acelerômetro, o contrário também é verdadeiro.

Para entender melhor este filtro segue um videozinho rápido: https://www.youtube.com/watch?v=whSw42XddsU
Recomenda-se também o estudo de angulos de euler e tilt usando aceleração, segue uma rápida referência:
Angulos de euler: https://www.youtube.com/watch?v=wg9bI8-Qx2Q
Extraindo ngulos da aceleração: https://www.nxp.com/docs/en/application-note/AN3461.pdf

Ajustes de filtro de kalman

O filtro de kalman, é um filtro mais genérico (se encaixa em diversas situações, sendo necessário uma modelagem), complexo e caro computacionalmente, sendo utilizado com mesmo objetivo do filtro complementar, porém, diferente do filtro complementar, ele consegue estimar de forma 'automática' o melhor valor de mu.

Os parâmetros deste filtro se encontram em 'Variável de ajuste do filtro de kalman', sendo eles vários, dentre eles os valores A,B e C que formam o modelo necessário para nossa fusão de dados.

> NOTE: o filtro de kalman utilizado foi implementado no modulo 'kalman.m' (mesmo já existindo um em bibliotecas nativas do matlab), porém de forma mais simples sem todos os parâmetros que o filtro suporta, para melhor entendimento do seu funcionamento.

Por fim, fica aqui duas breve fontes rápidas para melhor entendimento de como o filtro funciona:
Como o filtro funciona: https://www.youtube.com/watch?v=urhaoECmCQk
Modelagem do filtro para IMU:
https://www.researchgate.net/publication/261038357_Embedded_Kalman_Filter_for_Inertial_Measurement_Unit_IMU_on_the_ATMega8535

Ajustes de filtro madgwick

O filtro de madgwick é relativamente novo (2010) desenvolvido com o propósito de fundir os dados do MPU para obter uma melhor estimativa da posição de rotação (tilt) gastando o minimo de processamento, sendo este filtro possível de ser utilizado em sistemas embarcados para processamento e estimativa da posição de rotação em tempo real. Diferente dos demais filtros este realiza os calculos no plano de quarternios o que possibilitar que ele seja mais rápido e livre de gimbal lock, entretanto a conversão do quarternio para a representação em angulo de euler ainda esta sugeita a gimbal lock.

O algoritmo que implementa o filtro é bem simples, entretanto a matemática por trás não, portanto foi utilizado uma biblioteca de terceiros para realizar seu calculo e conversão para angulos de euler.

O filtro permite alterar o parâmetro bêta, que controla a velocidade de convergencia do algoritmo, disponível em 'Variável de ajuste do filtro madgwick'.

Para entender melhor o funcionamento do filtro, recomenda-se o estudo de quarternios e do filtor em si, abaixo segue algumas referências, rápidas:

Quaternios: https://www.youtube.com/watch?v=q-ESzg03mQc
Algoritmo usado: https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

Criação de mais janelas de plotagem

Como citado na seção de alteração de layout, também é possível criar outras janelas, utilizando do mesmo sistema, basta seguir os passos:

- 1: Criar um objeto render passando a frequência de renderização e a matriz de layout:

```
plot_1 = render(freq_render, layout);
```

- 2: Definir as propriedades (titlo, nome dos eixos, e nome das linhas):

```
plot_1.setProperties(Acel, 'Aceleração em g', 'Amostra', 'g', {'aX', 'aY', 'aZ'});
...
```

- 3: Definir quais variáveis são responsáveis por cada linha no plot:

```
plot_1.setSource(Acel, 'ax', 'ay', 'az');
...
```

- 4: Por fim você pode utilizar a função 'try_render()' para renderizar na frequencia definida ou 'force_render()' para renderizar agora.

> NOTE: os plots definidos seguem a ordem X,Y,Z, é obrigatório seguir a ordem, e passar todos eles. Não é possível alterar a cor deles. Note também que as função 'try_render()' somente irá renderizar se o periodo de plotagem já tiver passado, caso contrário ele não irá renderizar, sendo necessário chamar a função novamente (em loop).

Criação de novos items de layouts

Também é possível criar novos itens de layout, para isto basta criar 3 variáveis com o mesmo tamanho, não necessariamente precisa-se seguir o tamanho da janela de plotagem, uma vez que é plotado todos os dados contidos nas variáveis.

Após criar os 3 dados deve-se definir seu item nas macros (não é necessário, mas é recomendado, as funções do render entendem o item como string, definir uma variável para guardar esta string garante o uso correto dela, evitando escrita incorreta) e, adiciona-ló no layout desejado.

Por fim basta seguir os passos 2 e 3 em Criação de mais janelas de plotagem, utilizando o objeto da janela desejado para que ele seja renderizado.

Lembrando que cabe ao usuário certificar que a atualização dos dados das variáveis criadas ocorram corretamente, o objeto de renderização somente realiza a plotagem destas variáveis.

Análise de dados e exemplos

Para facilitar a compreenção dos dados, temos gravado pequenos vídeos rápidos apresentando e explicando os gráficos com algums movimentos caractetisticos:

Neste vídeo são tradados os movimentos de:

- 01 - Rotação em Y de 90º (Demonstração de gimbal lock) :
- 02 - Rotação em Z de 360º (Demonstração de angulo relativo) :
- 03 - Rotação em XYX sequencial de 90º (simulação de rotação em Z com gimbal lock) :
- 04 - Rotação em ZXZ sequencial de 90º (simulação de rotação em Y sem gimbal lock) :
- 05 - Rotação em XYZ sequencial de 45º :
- 06 - Rotação em XYZ individual de 90º :
- 07 - Rotação em XYZ sequencial de 90º :

Detalhes de implementação

Fluxograma

Segue o link para o fluxograma do programa principal: docs/main-flowchart.pdf

Kalman

Este modulo é uma classe que realiza o filtro de kalman em sua versão mais simples, criado para facilitar o entendimento do funcionamento do filtro, por mais que o matlab possue uma biblioteca nativa que realiza este calculo de forma ainda mais completa.

O modulo possui duas funções seguindo o modelo de kalman, de predição (que estima o valor da próxima amostra) e o de update (que realiza a correção da estimativa com base em amostas lidas).

Este módulo foi construido utilizando como base o conjunto de videos em: https://www.youtube.com/watch?v=urhaoECmCQk

Leitor

O modulo de leitor é uma classe que realiza leituras tanto via porta serial quando via arquivo. Ele possui 5 funções:

- **set_serial_reader:** Inicializa o cursor para leitura serial, e chama a função '*wait_start_signal*'
- **set_file_reader:** Inicializa o cursor para leitura via arquivo, e chama a função '*wait_start_signal*'
- **wait_start_signal:** Lê do meio selecionado e escreve na saida do console até encontrar os delimitadores de inicio
- **read_sample:** Lê do meio selecionado uma amostra dos sensores, se o dado estive no formato incorreto, ou for o fim retorna vazio
- **delete:** Fecha o cursor de leitura seja ela serial ou arquivo

Renderizador

O modulo renderizados pe uma classe que realiza a plotagem dos dados em diversos gráficos em uma mesma janela. Cada objeto desta classe criado é uma janela idependente, contando com as features de renderização em 'tempo real' e ajutes de layout.

Ao criar um objeto deste módulo, ele irá calcular as posições dos gráficos seguindo o layout escolhido, por fim abrindo e plotando os gráficos em uma janela.

> Vale resaltar que os gráficos são sempre de 3 eixos seguindo a ordem X, Y e Z

Os objetos deste modulo possuiem 4 funções:

- **setProperties:** Define propriedades de cada gráfico na janela (titulo, labels e nome das series(linhas X,Y e Z do gráfico))
- **setSource:** Define qual as variáveis que seram usadas em cada série do gráfico
- **try_render:** Realiza a atualização dos gráficos tentando respeitar a frequência definida, com a informação das variáveis. Somente executa se o tempo passado for maior ou igual ao periodo da frequencia
- **force_render:** Realiza a atualização dos gráficos imediatamente, independente da frequência definida
- **linkAllAxes:** Linka o eixo de todos os gráficos da janela, assim todo zoom aplicado a uma janela é aplicado a todas as outras

> NOTE: a função 'setSource' recebe as variáveis no tipo 'string' e não um ponteiro (padrão do matlab), por isso o algoritmo considera que estas variáveis estão localizadas no modulos executável (main, ou script, o arquivo que é executado). Não tente utilizar este modulo usando outro modulo que não seja o arquivo executado, pois as variáveis não vão estar no escopo.

Helpers

Na pasta helpers temos diversas funções, com funcionalidades diferentes, utilizado no script principal com o objetivo de deixar o código mais limpo e objetivo, para saber detalhes de cada fução verifique o arquivo da função em questão.

Referências

Matriz de rotação em 2d:

- https://www.youtube.com/watch?v=4srS0s1d9Yw

Matriz de rotação em 3d:

- https://www.youtube.com/watch?v=wg9bI8-Qx2Q

Angulos de Euler:

- https://www.youtube.com/watch?v=3Zjf95JEw2UVhttps://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf

Quaternions:

- https://www.youtube.com/watch?v=jlskQDR8-bY
- https://www.youtube.com/watch?v=zjMuIxRvygQ
- https://www.youtube.com/watch?v=d4EgbgTm0Bg
- https://www.youtube.com/watch?v=q-ESzg03mQc
- https://eater.net/quaternions/
- https://www.weizmann.ac.il/sci-tea/benari/sites/sci-tea.benari/files/uploads/softwareAndLearningMaterials/quaternion-tutorial-2-0-1.pdf

Extraindo angulos da aceleração:

- https://www.nxp.com/docs/en/application-note/AN3461.pdf

Unidade de medição inercial usando acelerometro, giroscopio e magnetometro:

- https://www.youtube.com/watch?v=T9jXoG0QYIA

Juntar acelerometro com gps para prever posição e velocidade:

- https://www.youtube.com/watch?v=6M6wSLD-8M8
- https://www.youtube.com/watch?v=RP7vkhDNNss
- https://www.youtube.com/watch?v=6qV3YjFppuc
- https://www.youtube.com/watch?v=0rIvvYgmTvI

Filtro Complementar:

- https://www.youtube.com/watch?v=whSw42XddsU

Filtro de Kalman:

- https://www.youtube.com/watch?v=mwn8xhgNpFY
- https://www.youtube.com/watch?v=urhaoECmCQk
- https://www.youtube.com/watch?v=Os6V1InUPZo
- https://www.researchgate.net/publication/261038357_Embedded_Kalman_Filter_for_Inertial_Measurement_Unit_IMU_on_the_ATMega8535

Filtro de Madgwick:

- tutorial resumido: https://nitinjsanket.github.io/tutorials/attitudeest/madgwick
- artigo do Madgwick: https://www.x-io.co.uk/res/doc/madgwick_internal_report.pdf
- Algoritmos: https://x-io.co.uk/open-source-imu-and-ahrs-algorithms/

Usando MPU9520 e plotando em 3D no matlab:

- https://www.peteletricaufu.com/static/ceel/artigos/artigo_269.pdf