

CENTRO UNIVERSITÁRIO FEI

FELIPE BASSI GUERREIRO

FELIPHE GONÇALVES GALIZA

LUIS FERNANDO MARTIN VILLANUEVA SILVA

NICOLE JOCYS

LAXIS: Dispositivo Autoestabilizador

São Bernardo do Campo

2016

CENTRO UNIVERSITÁRIO FEI

FELIPE BASSI GUERREIRO

FELIPHE GONÇALVES GALIZA

LUIS FERNANDO MARTIN VILLANUEVA SILVA

NICOLE JOCYS

LAXIS: Dispositivo Autoestabilizador

Trabalho de Conclusão de Curso, apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia de Automação e Controle. Orientado pelo Prof. Dr. Mario Kawano.

São Bernardo do Campo

2016

LAXIS : dispositivo autoestabilizador / Felipe Bassi Guerreiro...[et al.].
São Bernardo do Campo, 2016.

81 f. : il.

Trabalho de Conclusão de Curso - Centro Universitário FEI.

Orientador: Prof. Dr. Mario Kawano.

1. Gimbal. 2. IMU. 3. Brushless. I. Bassi Guerreiro, Felipe. II.
Gonçalves Galiza, Felipe. III. Martin Villanueva Silva, Luis Fernando.
IV. Jocys, Nicole. V. Kawano, Mario, orient. VI. Título.

Elaborada pelo sistema de geração automática de ficha catalográfica da FEI com os
dados fornecidos pelo(a) autor(a).

FELIPE BASSI GUERREIRO
FELIPHE GONÇALVES GALIZA
LUIS FERNANDO MARTIN VILLANUEVA SILVA
NICOLE JOCYS

LAXIS: Dispositivo Autoestabilizador

Trabalho de Conclusão de Curso, apresentado ao Centro Universitário FEI, como parte dos requisitos necessários para obtenção do título de Bacharel em Engenharia de Automação e Controle.

Comissão julgadora

Orientador: Prof. Dr. Mario Kawano

Prof. Dr. Aldo Artur Belardi

Prof. Dr. Renato Camargo Giacomini

São Bernardo do Campo

2016

Dedicamos este trabalho a nossos pais, amigos
e a todos aqueles que nos motivaram a lutar
pelo que acreditamos.

AGRADECIMENTOS

Ao nosso orientador, Prof. Dr. Mario Kawano, pelo apoio que nos deu, por todos os ensinamentos e pela ajuda nas metas e objetivos necessários para o desenvolvimento deste trabalho.

Aos professores Pedro Luiz Benko, Marco Antonio Assis de Melo e Reinaldo Augusto da C. Bianchi, que foram bastante atenciosos com ideias para escolha do tema e sanaram todas as dúvidas que tivemos para dar continuidade a elaboração do projeto.

Aos nossos colegas Danilo Pucci e Sheldon Zabulon, pelo tempo dedicado a nos auxiliar com os seus conhecimentos, opiniões e experiências.

Aos funcionários do departamento de elétrica: Acácio, Ari, Douglas e Yuri, por toda ajuda e disponibilidade de equipamentos para realização do projeto.

A toda equipe da empresa Xfly Brasil, pelo apoio, atenção e recepção que foram fundamentais para dar prosseguimento na finalização do projeto.

A toda equipe do Robô FEI, pela instrução de que tipos de materiais deveríamos comprar e pelas dicas de como utilizá-los.

A todos os familiares que sempre nos apoiaram e nos incentivaram a estudar para alcançar nossos objetivos.

Aos nossos namorados e namoradas, pelo incentivo e pelos momentos de descontração e dedicação.

Ao Centro Universitário FEI, por toda infraestrutura e disponibilidade de recursos essenciais no desenvolvimento deste trabalho.

“Eu odiava cada minuto de treino, mas eu dizia: *Não desista. Sofra agora e viva o resto da sua vida como campeão.*”

Muhammad Ali

RESUMO

Inspirado pelo avanço da tecnologia e a vasta utilização do conceito de dispositivos denominados “Gimbal” em iniciativas militares, motores de foguetes, navios, submarinos, aviões, cronômetros marítimos, fotografias e filmagens profissionais, surgiu-se o interesse por parte do grupo em desenvolver um dispositivo capaz de estabilizar a orientação de objetos dinamicamente, com foco em melhorar a qualidade de vídeos gravados por câmeras e smartphones. Motores fixados à estrutura da câmera controlam continuamente os três eixos ortogonais de rotação, corrigindo as variações de sua posição, mantendo a imagem praticamente imóvel. Atualmente esta tecnologia vem sendo aprimorada e seu custo diminuído, tornando-se um produto cada vez mais acessível para outras finalidades.

Para o funcionamento destes dispositivos é essencial conhecer a orientação da câmera nos seus 3 eixos, os quais podem ser obtidos através do uso de sensores como acelerômetros e giroscópios. Portanto, neste projeto emprega-se o uso de um sensor do tipo IMU (Inertial Measurement Unit), onde um microcontrolador envia comandos para a leitura da orientação final do sistema e atua no sinal de entrada de três motores brushless. A estratégia de controle empregada no desenvolvimento deste projeto trata-se de um Controlador PID implementado em linguagem C, possibilitando também a utilização futura deste projeto como plataforma de pesquisa e teste de outros métodos de Controle.

Palavras-chave: Estabilizador, Gimbal, IMU, acelerômetro, giroscópio, motor sem escovas

ABSTRACT

Inspired by the advancement of technology and the widespread use of the concept of devices called "Gimbal" in military initiatives, rocket engines, ships, submarines, aircraft, photographs and professional filming, the group grew an interest in developing a device capable of stabilizing objects' orientation in a dynamically way - focusing on improving the quality of videos recorded by cameras and smartphones. Motors attached to the camera structure continuously control the rotation of three orthogonal axes, correcting the position variations and maintaining an immobile image. This technology is being currently improved - and its cost reduced - making it a product increasingly accessible for other purposes.

For the operation of these devices is essential to know the camera orientation in its three axes, which can be obtained through the use of sensors such as accelerometers and gyroscopes. This project uses an IMU (Inertial Measurement Unit) sensor, in which a microcontroller determines the final orientation of the system and sends distinct signals to the three brushless motors. The control strategy used in the development of this project is a PID controller implemented in C language, thus enabling the use of this project as a platform for research and testing other control methods.

Keywords: Stabilizer, Gimbal, IMU, accelerometer, gyroscope, brushless motor

LISTA DE ILUSTRAÇÕES

Figura 1: Tanque de Guerra	14
Figura 2: Estabilizador de arma	15
Figura 3: Sistema de controle malha aberta.....	19
Figura 4: Sistema de Controle Malha Fechada.....	19
Figura 5: Comparação entre Sistema de Loop aberto e Sistema de Loop fechado	20
Figura 6: Sistema de controle feedback	20
Figura 7: Diagrama de bloco com os parâmetros do controle por feedback	21
Figura 8: Primeiro método de Ziegler e Nichols	25
Figura 9: Segundo método de Ziegler e Nichols	26
Figura 10: Representação de Rotações em um espaço 3D	27
Figura 11: Yaw, Pitch e Roll	27
Figura 12: Estrutura de um acelerômetro de dois eixos	30
Figura 13: Estrutura de um giroscópio de um eixo.....	31
Figura 14: IMU multiaxial baseado na abordagem "folded MEMS"	32
Figura 15: Unidades inerciais	33
Figura 16: MPU-6050.....	33
Figura 17: Eixos IMU MPU-6050.....	34
Figura 18: Forças acelerômetro	34
Figura 19: Forças no giroscópio	35
Figura 20: Fusão de Sensores	37
Figura 21: Motor DC com escovas.....	38
Figura 22: Motor brushless	39
Figura 23: Motor GBM2208H-80T	39
Figura 24: Modelo dinâmico do atuador	41
Figura 25: Componente L6234 operando em altas velocidades	42
Figura 26: L6234 operando em baixa velocidade (Controle de Posição).....	43
Figura 27: Diagrama de Blocos	44
Figura 28: Circuito montado em Protoboard	45
Figura 29: Placa BaseCam.....	46
Figura 30: Arquitetura ARM Cortex M4.....	47
Figura 31: Kit L6234	49
Figura 32: Montagem placa didática	50

Figura 33: Localização IMU.....	50
Figura 34: Tela de configurações.....	51
Figura 35: SetPoint	52
Figura 36: Controle PID	52
Figura 37: Parâmetros garrafa pet.....	55
Figura 38: Posição das forças na estrutura	56
Figura 39: Impressora Makerbot 2X.....	57
Figura 40: Configuração do dispositivo	58
Figura 41: Extrusion Speeds	58
Figura 42: Infill (densidade)	59
Figura 43: Propriedades do modelo	59
Figura 44: Multi-Material Printing	60
Figura 45: Raft.....	60
Figura 46: Supports and Bridging.....	61

LISTA DE TABELAS

Tabela 1: Características dos parâmetros de um controlador PID.....	23
Tabela 2: Primeiro Método de Ziegler e Nichols	26
Tabela 3: Segundo Método de Ziegler e Nichols	26
Tabela 4: Parâmetros iniciais do PID	53
Tabela 5: Segunda tentativa de parâmetros	54
Tabela 6: Parâmetros finais	54
Tabela 7: Parâmetros garrafa pet	55

LISTA DE ABREVIATURAS E SIGLAS

IMU	Inertial Measurement Unit
FEI	Fundação Educacional Inaciana Pe. Sabóia de Medeiros
URSS	União das Repúblicas Socialistas Soviéticas
INS	Inertial Navigation System
BFV	Bradley Fighting Vehicle
PID	Proporcional Integral Derivativo
PLC	Controlador Lógico Programável
MEMS	Múltiplas Entradas Múltiplas Saídas
BLDC	Brushless Direct Current
SPWM	Sinusoidal Pulse Width Modulation

SUMÁRIO

1. INTRODUÇÃO	14
2. DESENVOLVIMENTO.....	16
2.1 SISTEMAS DE CONTROLE	16
2.1.1 Características dos Sistemas de Controle.....	16
2.1.2 Requerimentos para um bom Sistema de Controle.....	16
2.1.3 Tipos de Sistemas de Controle.....	17
2.1.3.1 Sistema de controle de Circuito Aberto.....	18
2.1.3.2 Sistema de controle de Circuito Fechado	19
2.1.3.3 Feedback do Sistema de Controle de Loop	20
2.2 CONTROLADOR PID.....	22
2.3 MÉTODO DE ZIEGLER E NICHOLS	25
2.4 CINEMÁTICA E DINÂMICA DO DISPOSITIVO	27
2.4.1 Representação de rotação em três dimensões	27
2.4.2 Matriz Jacobiana, Velocidade e Torque.....	28
2.5 SENSORES INERCIAIS.....	30
2.5.1 Acelerômetros	30
2.5.2 Giroscópios	31
2.5.3 IMU	32
2.5.3.1 IMU MPU-6050	33
2.5.3.2 Cálculo da orientação a partir de dados do acelerômetro.....	34
2.5.3.3 Cálculo da orientação a partir de dados do giroscópio	35
2.5.3.4 Algoritmos de Fusão	36
2.6 MOTORES COM E SEM ESCOVAS	37
2.6.1 Motor escovado DC	37
2.6.2 Motor sem escovas (brushless)	38

2.6.3 Motor brushless GBM2208H-80T	39
2.6.4 Atuador dinâmico	40
2.7 SPWM.....	42
2.8 DIAGRAMA DE BLOCOS	44
2.9 PRIMEIRA EXPERIÊNCIA	44
2.9 PLACA MICROPROCESSADORA GIMBAL	45
2.9.1 Microcontrolador	47
2.9.2 Driver de Motor Trifásico	48
2.9.3 Conexão da placa microprocessadora	50
2.9.4 Sintonização do PID	52
2.10 ESTRUTURA.....	55
2.10.1 Determinação da estrutura e posicionamento dos eixos	55
2.10.2 Impressora 3D	56
<i>2.10.2.1 Configurações de impressão</i>	58
3. CONCLUSÃO.....	62
REFERÊNCIAS BIBLIOGRÁFICAS	63
APÊNDICE A – Código em C	65
APÊNDICE B – Design Estrutura	75

1. INTRODUÇÃO

O conceito de estabilizador começou a ser utilizado na década de 30, pelos soviéticos na antiga URSS durante a Segunda Guerra Mundial. Mas o primeiro estabilizador de arma eficaz foi construído apenas em 1944, pela empresa americana BAE Systems Land and Armaments, e foi denominado Bradley Fighting Vehicle em homenagem ao General Omar Bradley. Considerando que o problema consistia em tentar manter a arma dos tanques de guerra em um alvo enquanto o tanque balançava e sacudia através de um campo de batalha áspero.



Figura 1: Tanque de Guerra

Fonte: <<https://s-media-cache-ak0.pinimg.com/736x/8d/5f/64/8d5f6442115cbb7c4ee92d2f90eb88b6.jpg>>
Acesso em Jun. 2016

Um estabilizador de arma é um dispositivo destinado a facilitar a apontar uma arma, compensando para o movimento da plataforma na qual a arma está montada.

O mecanismo geralmente inclui um dispositivo de referência angular, como giroscópios e servo. No caso de um tanque, ele possui um servo para estabilização da torreta e outro para a elevação da arma. A pontaria é então feita pela entrada de controle do mecanismo, em vez de diretamente sobre a arma. O mecanismo de controle geralmente tem outras funções, tais como a aplicação de super-elevação e conduzindo o alvo de acordo com a sua velocidade, tornando-se um sistema de controle de fogo, e algumas armas são totalmente automáticas.



Figura 2: Estabilizador de arma

Fonte: Autor "adaptado de" Popular Science magazine, 1944.

O foco deste trabalho é a estabilização de objetos dinamicamente, baseando-se nos conceitos de Controle, e que seja capaz de regular a orientação de dispositivos por meio de giroscópios e acelerômetros.

O termo "Gimbal" tem o significado de um suporte oscilante que permite a rotação de um objeto sobre o mesmo eixo. É um mecanismo com recurso de motores que mantém um objeto perfeitamente nivelado no plano horizontal e sem vibrações, mesmo que o dispositivo incline conforme a mudança de posição. Por exemplo, em um navio, os giroscópios, bússolas a bordo, fogões e até mesmo suporte de bebidas normalmente usam giroscópios para mantê-los na posição vertical com relação ao horizonte, apesar do balanço do navio.

Existem gimbals com 2 eixos que compensam tanto o eixo horizontal como vertical, e recentemente surgiram os de 3 eixos que além dos outros dois, ainda compensam o eixo do leme do modelo, fazendo com que as filmagens fiquem muito melhores.

Algumas aplicações deste mecanismo são citadas abaixo:

Navios e submarinos;

Aeronaves com o sistema INS;

Fotografias e imagens;

Filmagens e vídeos.

1. DESENVOLVIMENTO

2.1 SISTEMAS DE CONTROLE

Um sistema de controle é um sistema de dispositivos ou conjunto de dispositivos, que gera comandos e dirige ou regula o comportamento de outro dispositivo ou sistema para conseguir determinados resultados. Em outras palavras, a definição de sistema de controle pode ser reescrita como um sistema de controle que controla outro sistema.

A medida que a civilização humana está sendo continuamente modernizada, a demanda da automação aumenta concomitantemente. Automação requer precisos controles de dispositivos. Nos últimos anos, os sistemas de controle desempenharam um papel primordial no desenvolvimento e avanço da tecnologia e da civilização moderna. Praticamente todos os aspectos da nossa vida do dia-a-dia é afetada mais ou menos por algum sistema de controle. Geladeiras, freezers, ar condicionados, ferros automáticos, automóveis, todos são sistema de controle. Estes sistemas também são utilizados em processos industriais de potência. Encontramos sistema de controle no controle de produtos, sistemas de armas, sistemas de transporte, sistema de energia, tecnologia espacial, robótica e muito mais qualidade. Os princípios da teoria de controle são aplicáveis aos campos da engenharia e não engenharia.

2.1.1 Características dos Sistemas de Controle

A característica principal do sistema de controle é que deve haver uma relação matemática clara entre a entrada e a saída do sistema. Quando a relação entre a entrada e a saída do sistema pode ser representada por uma proporcionalidade linear, o sistema é chamado sistema de controle linear. Mais uma vez, quando a relação entre a entrada e a saída não pode ser representado pela proporcionalidade linear simples, em vez de entrada e de saída estão relacionadas por uma relação não linear, o sistema é referido como sistema de controle não-linear.

2.1.2 Requerimentos para um bom Sistema de Controle

Precisão: A precisão é a tolerância de medição do instrumento e define os limites dos erros cometidos quando o instrumento é usado em condições normais de operação. A precisão pode ser melhorada pela utilização de elementos de retroação. Para aumentar a precisão de todo o sistema de controle, o detector de erro deve estar presente no sistema de controle.

Sensibilidade: Os parâmetros do sistema de controle estão sempre mudando com a mudança das condições circundantes, distúrbios internos ou quaisquer outros parâmetros. Esta mudança pode ser expressa em termos de sensibilidade. Qualquer sistema de controle deve ser insensível a estes parâmetros, mas deve ser apenas sensível a sinais de entrada.

Ruído: Um sinal de entrada indesejada é conhecido como ruído. Um bom sistema de controle deve ser capaz de reduzir o efeito de ruído para um melhor desempenho.

Estabilidade: É uma característica importante do sistema de controle. Para o sinal de entrada limitada, a saída tem de ser limitada, e se a entrada for zero, então o produto tem de ser igual a zero, em seguida, um tal sistema de controle é dito ser um sistema estável.

Largura de banda: Uma faixa de frequência operacional decide a largura de banda do sistema de controle. A largura de banda deve ser grande quanto possível para a resposta de um bom sistema de controle de frequência.

Velocidade: É o tempo tomado pelo sistema de controle para alcançar a sua saída estável. Um bom sistema de controle processa em alta velocidade. O período transitório para tal sistema é muito pequeno.

Oscilação: um pequeno número de oscilação constante ou oscilação de saída do sistema tendem a ser estáveis.

2.1.3 Tipos de Sistemas de Controle

Existem vários tipos de sistemas de controle, mas todos eles são criados para controlar as saídas. O sistema usado para controlar a posição, velocidade, aceleração, temperatura, pressão, tensão e corrente etc são exemplos de sistemas de controle.

Tomemos um exemplo de controlador de temperatura simples de uma sala para deixar claro o conceito. Suponhamos que existe um elemento de aquecimento simples, que é aquecido, desde que o fornecimento de energia elétrica esteja ligado. Enquanto o interruptor do aquecedor de alimentação de alimentação estiver ligado a temperatura ambiente aumenta, e após alcançar a temperatura desejada a fonte de alimentação é desligada. Mais uma vez, devido a temperatura ambiente, a temperatura diminui e, em seguida, manualmente o elemento de aquecimento é ligado até a atingir a temperatura ambiente desejada novamente.

Desta forma, pode-se controlar manualmente a temperatura ambiente ao nível desejado. Este é um exemplo de sistema de controle manual. Este sistema pode ser ainda melhorado utilizando o arranjo de comutação temporizador da fonte de alimentação, onde o fornecimento ao elemento de aquecimento é ligado e desligado em intervalos predeterminados para atingir o nível desejado de temperatura do ambiente. Há uma outra forma melhorada de controlar a temperatura do ambiente. Nele, um sensor mede a diferença entre a temperatura real e temperatura desejada. Se há alguma diferença entre eles, as funções de elementos de aquecimento reduzem a diferença e quando esta diferença se torna menor do que o nível predeterminado, os elementos de aquecimento param de funcionar. Ambas as formas do sistema são sistemas de controle automático.

No anterior, uma entrada do sistema é inteiramente independente da saída do sistema. A temperatura do ambiente (saída) aumenta a medida em que o interruptor da fonte de alimentação é mantido ligado. Isso significa que o elemento de aquecimento produz calor, uma vez que a fonte de alimentação é mantida ligada e a temperatura final da sala não tem controle algum para alimentar o sistema. Este sistema é referido como sistema de controle de malha aberta.

Mas neste último caso, os elementos de aquecimento da função do sistema, dependem da diferença entre a temperatura real e temperatura desejada. Esta diferença é denominada como erro do sistema. Este sinal de erro é alimentado de volta ao sistema para controlar a entrada. A medida em que o sinal de entrada para saída e o feedback de erro criam um circuito fechado, este tipo de sistema de controle é referido como sistema de controle de circuito fechado.

Assim, existem dois tipos principais de sistemas de controle. Eles são os seguintes:

Sistema de controle de circuito aberto

Sistema de controle de circuito fechado

2.1.3.1 Sistema de controle de Circuito Aberto

Um sistema de controle em que a ação de controle é totalmente independente da saída do sistema, então, ele é chamado de sistema de controle malha aberta. O Sistema de controle manual é também um sistema de controle malha aberta. A figura abaixo mostra o diagrama de blocos do sistema de controle de circuito aberto, no qual a saída de processo é totalmente independente da ação do controlador.

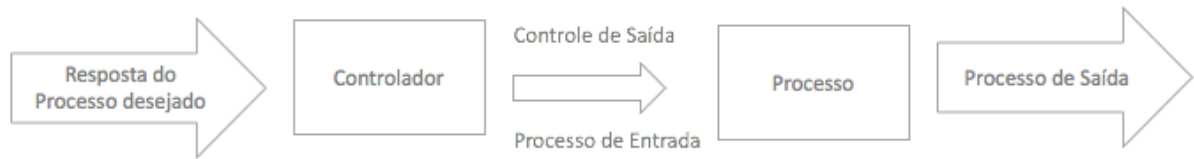


Figura 3: Sistema de controle malha aberta

Fonte: < <http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/>> Acesso em: Jun. 2016

2.1.3.2 Sistema de controle de Circuito Fechado

Sistema de controle em que a saída tem um efeito sobre a quantidade de entrada de tal maneira que a quantidade de entrada ajusta-se com base na saída gerada. É então chamado sistema de controle de circuito fechado. O sistema de controle loop aberto pode ser convertido em sistema de controle circuito fechado, fornecendo um feedback. Esse feedback faz automaticamente as alterações adequadas na produção devido à perturbação externa. Neste sistema de controle de circuito fechado é chamado de sistema de controle automático. A figura abaixo mostra o diagrama de blocos do sistema de controle de circuito fechado em que a realimentação é tomada a partir da saída e alimentado para entrada.

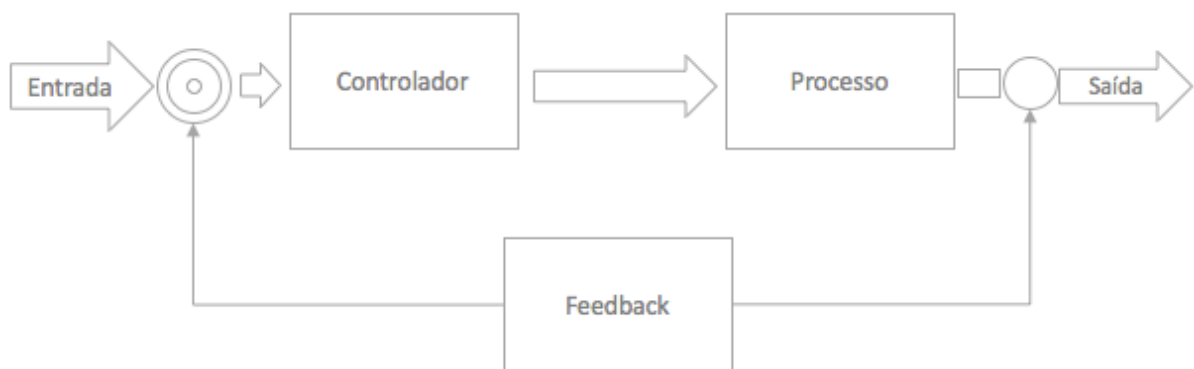


Figura 4: Sistema de Controle Malha Fechada

Fonte: < <http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/>> Acesso em: Jun. 2016

Sistema de Loop Aberto	Sistema de Loop Fechado
1 O elemento de feedback está ausente	O elemento de feedback está sempre presente
2 O detector de erro não está presente	O detector de erro está sempre presente
3 Este é Estável	Pode se tornar instável
4 Fácil construção	Construção complicada
5 Custo econômico	Alto custo
6 Pequena largura de banda	Grande largura de banda
7 Impreciso	Preciso
8 Pouca manutenção necessária	Muita manutenção necessária
9 Menos confiável	Mais confiável

Figura 5: Comparação entre Sistema de Loop aberto e Sistema de Loop fechado

Fonte: < <http://www.electrical4u.com/> > Acesso em: Jun. 2016

2.1.3.3 Feedback do Sistema de Controle de Loop

Um feedback é uma ferramenta comum e poderosa na concepção de um sistema de controle. Feedback do circuito é a ferramenta que leva a saída do sistema em consideração e permite que o sistema ajustar seu desempenho para atender a um resultado desejado do sistema. Em qualquer sistema de controle, a saída é afetada devido a mudança de condição ambiental ou qualquer tipo de perturbação. Então, um sinal é tomado a partir da saída e é alimentado de volta para a entrada. Este sinal é comparado com entrada de referência e, em seguida um sinal de erro é gerado. Este sinal de erro é aplicado ao controlador e a saída é corrigida. Tal sistema é chamado sistema de feedback. A figura abaixo mostra o diagrama de blocos do sistema de feedback.

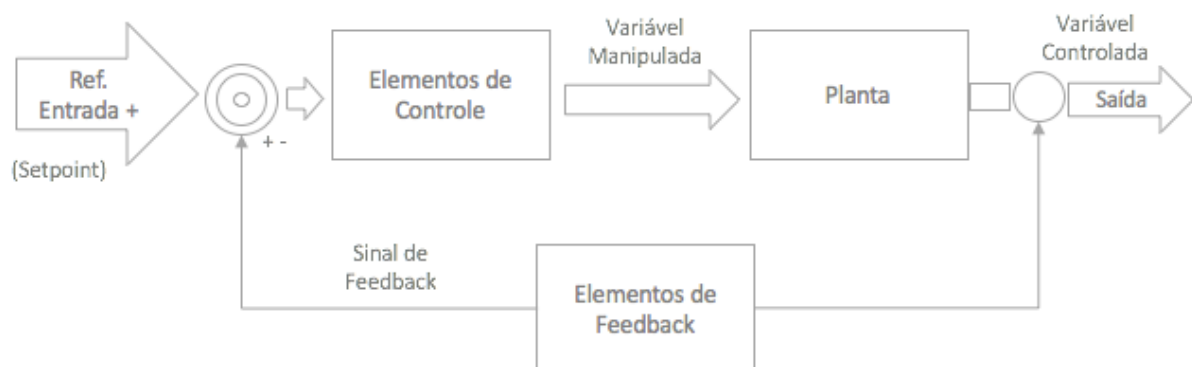


Figura 6: Sistema de controle feedback

Fonte: < <http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/> > Acesso em Jun. 2016

Quando o sinal de feedback é positivo, então o sistema é chamado sistema de feedback positivo. Para o sistema de feedback positivo, o sinal de erro é a adição de sinal de entrada de referência e sinal de feedback. Quando o sinal de feedback for negativo então o sistema é chamado de sistema de feedback negativo. Para o sistema de feedback negativo, o sinal de erro é dado pela diferença de sinal de entrada de referência e sinal de feedback.

Na figura abaixo, pode-se verificar a representação do sistema de feedback, onde:

R = Sinal de entrada

E = Sinal de erro

G = Ganho caminho a seguir

H = O feedback

C = O sinal de saída

B = Sinal de feedback

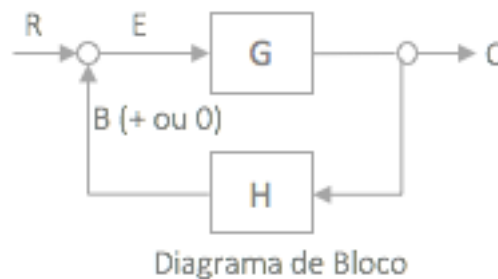


Figura 7: Diagrama de bloco com os parâmetros do controle por feedback

Fonte: < <http://www.electrical4u.com/control-system-closed-loop-open-loop-control-system/> > Acesso em: Jun. 2016

Erro entre a entrada e saída do sistema do sistema é reduzida;

Ganho do sistema é reduzido por um fator de $1 / (1 \pm GH)$;

Melhora na sensibilidade;

A estabilidade pode ser afectada;

Melhorar a velocidade de resposta.

2.2 CONTROLADOR PID

O controlador PID é o mais comum e útil sistema de controle na engenharia. Na maioria dos casos, os feedback loops são controlados utilizando o algoritmo PID. A principal razão pela qual o feedback é muito importante em sistemas de controle, é a de ser capaz de atingir um ponto de ajuste independentemente de perturbações ou qualquer variação nas características de qualquer formato.

Este controlador é sempre projetado para corrigir o erro entre medidas de valores de processo e um set-point desejado em um sistema. O controlador fornece a excitação necessária pelo sistema e que se destina a controlar o seu comportamento. O controlador PID possui várias categorias de arranjos estruturais, a maioria comum destes são as séries e as estruturas paralelas e, em alguns casos, existem a forma híbrida da série e as estruturas paralelas.

Existem algumas ações de controle que podem ser adquiridas através de qualquer um dos dois parâmetros do controlador PID. Dois parâmetros podem funcionar enquanto mantém o terceiro em zero. Então controlador PID torna-se, por vezes, PI (proporcional integral), PD (proporcional derivativo) ou mesmo P ou I. O termo D derivativo é responsável pela medição de ruído, enquanto o termo integral é destinado a atingir o valor alvo do sistema. Muitos sistemas eletrônicos complexos são fornecidos com malha de controle PID. Nos dias modernos controladores PID são utilizados em PLC na indústria. Ele é aplicável a muitas ações de controle, mas não tem um bom desempenho em caso de controle ideal. A principal desvantagem é o caminho feedback. O PID não é fornecido em qualquer modelo do processo. Outra desvantagem deste controlador é ser um sistema linear onde a parte derivativa é sensível a ruídos, e uma pequena quantidade de ruído pode causar grande mudança na saída. O controle PID combina vantagens das ações de controle proporcionais, derivativos e integrais.

Na imagem abaixo é ilustrado um diagrama típico de um controlador PID comum. Tipicamente, a função mostrada na equação é aplicável neste tipo de projeto.

$$K_P + \frac{K_I}{s} + K_D \cdot s = \frac{K_D s^2 + K_P s + K_I}{s}$$

Onde:

K_p = Ganho proporcional

K_i = Ganho Integral

K_d = Ganho derivativo

A diferença entre o valor de entrada desejado e a saída atual é representado pelo erro amostral. Em malha fechada, o erro é mandado para o controlador e este irá realizar a computação integrativa e derivativa do sinal de erro. Após isto, o sinal u (esforço de controle) que é a saída do controlador é agora igual a soma do produto do ganho proporcional K_p e a magnitude do erro, o produto do ganho integrativo K_i e a integral do erro e o produto do ganho derivativo K_d e a derivada do erro. Como demonstrado na equação a seguir:

$$u = K_p e + K_i \int e dt + K_d \frac{de}{dt}$$

O valor do sinal u é enviado continuamente para a planta com toda nova correspondente saída, sendo esta obtida com a continuidade do processo. A saída é enviada de volta a um novo sinal de erro, onde este é encontrado e o processo continua a se repetir.

O ganho proporcional K_p reduzirá o tempo de subida e deve reduzir ou remover o erro estacionário do sistema. O ganho integral K_i eliminará o erro estacionário, porém ocasiona um efeito negativo na resposta do transitório (uma pior resposta deve ser produzida neste caso). E o ganho derivativo K_d tenderá a aumentar a estabilidade do sistema, reduzindo a porcentagem de overshoot e melhorando a resposta do transitório do sistema. A tabela abaixo demonstra os efeitos na saída da planta, de forma compreensível, no incremento do ganho de cada controlador em um típico sistema de malha fechada.

Controlador	Tempo de subida	Tempo de acomodação	Overshoot	Erro estacionário
K_p	Diminui	Pequena mudança	Aumenta	Diminui
K_d	Pequena Mudança	Diminui	Diminui	Não muda
K_i	Diminui	Aumenta	Aumenta	Elimina

Tabela 1: Características dos parâmetros de um controlador PID

Fonte: Autor “adaptado de” Controle Avançado 2

A habilidade de misturar estes três parâmetros resultará em um sistema muito estável e eficiente. Deve ser notado que a relação entre os parâmetros dos três controladores pode não ser adquirida de forma exata por causa da interdependência entre eles. Apesar disso, é possível computar parâmetros particulares, onde os efeitos seriam notados com outros dois.

Para projetar um controlador PID é requerido alguns passos para obter um sistema mais eficaz e estável:

- 1) É aconselhável obter uma resposta em malha-aberta do sistema primeiro e subsequentemente determinar o que melhorar;
- 2) Adicionar um ganho proporcional para melhorar o tempo de subida;
- 3) Então, adicionar o ganho derivativo para melhorar a porcentagem de overshoot;
- 4) E talvez, adicionar o controle integrativo para eliminar o erro estacionário;
- 5) Após isso, ajustar cada um dos parâmetros deve ser importante para atingir um desempenho desejado na saída.

Também é importante apontar que nem sempre é necessário utilizar todos os parâmetros do PID. Na maioria dos casos com o controle do tipo PI, apenas com os ganhos proporcionais e integrativos, é capaz de controlar o sistema de forma eficiente.

Além disso, deve ser notado que o principal objetivo dos parâmetros PID é obter um rápido tempo de subida com o mínimo de overshoot e quase nada de erro de acomodação.

O objetivo do projeto do controlador PID é aproveitar as características particulares de cada uma dessas ações afim de se obter uma melhoria significativa do comportamento transitório e em regime permanente do sistema controlado.

Se um modelo matemático da planta pode ser obtido, então é possível aplicar várias técnicas de projeto na determinação dos parâmetros do controlador que vão impor especificações do regime transitório e do regime permanente do sistema de malha fechada. No entanto, nem sempre é possível obter o modelo matemático da planta tão facilmente, impossibilitando obter os parâmetros do PID de forma analítica. Portanto, para estes casos é aplicada uma abordagem experimental para ajustes do controlador PID.

2.3 MÉTODO DE ZIEGLER E NICHOLS

O processo de selecionar os parâmetros do controlador que garantam uma dada especificação de desempenho é conhecido como sintonia do controlador. Existem diversos métodos de sintonia do PID. Um dos mais famosos é o método de Ziegler e Nichols.

O método de Ziegler e Nichols sugerem valores de ganhos para cada ação (proporcional, derivativa e integrativa que fornecerão uma operação estável do sistema. Em outras palavras, as estimativas dos parâmetros do PID proporcionam um ponto de partida para o projeto, para que assim uma sintonia fina seja realizada para que as especificações do modelo transitório e permanente de malha fechada sejam atendidas.

Os métodos de Ziegler e Nichols são baseados em regras para determinar os parâmetros K_c , T_i e T_d da função de transferência do PID como mostrado abaixo:

$$u(t) = K_c[e(t) + \frac{1}{T_i} \int_0^t e(\tau) d\tau + T_d de(t) dt]$$

O primeiro método é aplicável a planta que não contem integradores, pólos complexos conjugados dominantes, por conta disso, geralmente, a curva de resposta ao degrau unitário pode ter aspecto de um S como mostrado na figura abaixo.

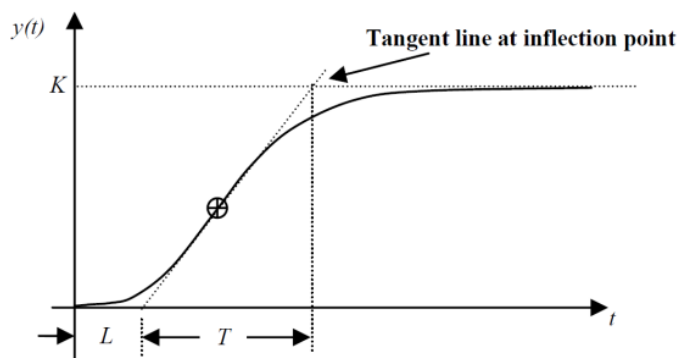


Figura 8: Primeiro método de Ziegler e Nichols

Fonte: < <http://www.electrical4u.com/nichols-plot/> > Acesso em: Jun. 2016

A curva com formato em S pode ser caracterizada através de duas constantes, o atraso L e a constante de tempo T . O atraso e a constante de tempo são determinados desenhando-se uma linha tangente no ponto de inflexão da curva com formato de S e determinando-se a interseção da linha tangente com o eixo dos tempos e linha $c(t) = K$, sendo K o valor estacionário da resposta ao degrau, como mostrado na figura a cima.

Ziegler e Nichols sugeriram para fixar os valores de K_c , T_i e T_d de acordo com a fórmula mostrada na tabela abaixo.

	K_c	T_i	T_d
P	T/L	∞	0
PI	$0,9 \cdot T/L$	$L/0,3$	0
PID	$1,2 \cdot T/L$	$2L$	$0,5L$

Tabela 2: Primeiro Método de Ziegler e Nichols

Fonte: Autor “adaptado de” Controle Avançado 2

No segundo método utiliza-se, primeiramente, a ação proporcional, e aumentando o K_c até que a saída passe a exibir, pela primeira vez, uma oscilação sustentada. Se a saída não exibir uma oscilação sustentável para nenhum K_c pode se assumir que esse método não se aplica.

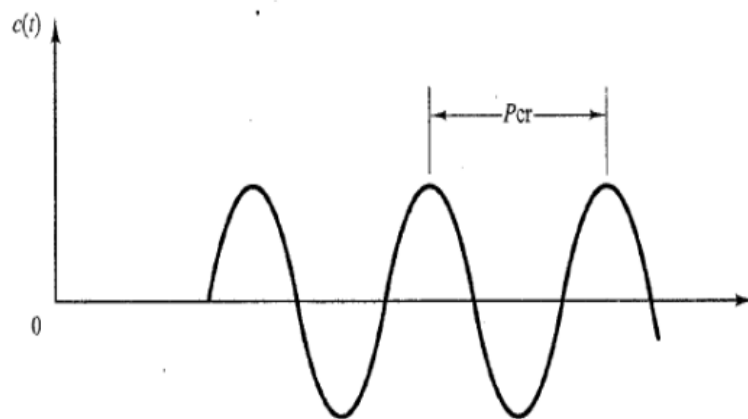


Figura 9: Segundo método de Ziegler e Nichols

Fonte: < <http://www.electrical4u.com/nichols-plot/> > Acesso em Jun. 2016

No caso de o sistema apresentar uma oscilação sustentável, os parâmetros do PID são determinados conforme a tabela abaixo. Onde K_{cr} é o valor de ganho proporcional em que o sistema primeiro apresenta uma oscilação sustentável e P_{cr} é o período de oscilações como mostrado a cima.

	K_c	T_i	T_d
P	$0,5 \cdot K_{cr}$	∞	0
PI	$0,45 \cdot K_{cr}$	$P_{cr}/1,2$	0
PID	$0,6 \cdot K_{cr}$	$P_{cr} \cdot 0,5$	$0,125 P_{cr}$

Tabela 3: Segundo Método de Ziegler e Nichols

Fonte: Autor “adaptado de” Controle Avançado 2

2.4 CINEMÁTICA E DINÂMICA DO DISPOSITIVO

2.4.1 Representação de rotação em três dimensões

Qualquer possível rotação de um objeto em um espaço 3D pode ser representada por rotações sucessivas em torno de três eixos ortogonais (Fig. 10). Neste projeto adotou-se a mesma terminologia originada pelo setor de aviação, onde estas rotações serão referenciadas como yaw, pitch e roll (Fig. 11).

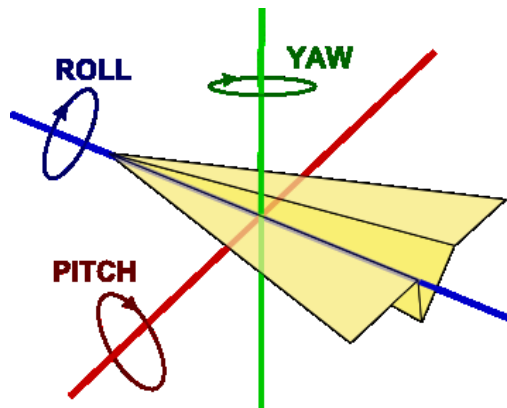


Figura 10: Representação de Rotações em um espaço 3D

Fonte: <<http://www.xojo3d.com/tut012.php>> Acesso em Jun. 2016

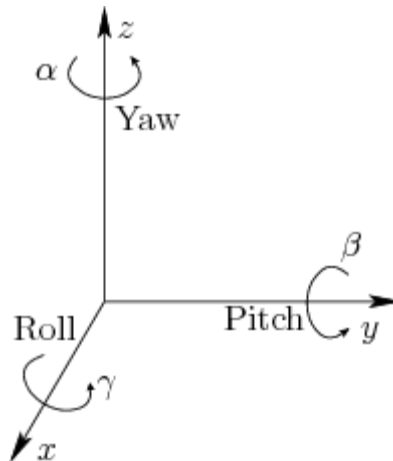


Figura 11: Yaw, Pitch e Roll

Fonte: <<http://www.xojo3d.com/tut012.php>> Acesso em Jun. 2016

1) O yaw é uma rotação no sentido anti-horário, no valor de α e em torno do eixo z. A matrix de rotação é dada por:

$$R_z(\alpha) = \begin{pmatrix} \cos \alpha & -\sin \alpha & 0 \\ \sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 1 \end{pmatrix}.$$

2) O pitch é uma rotação no sentido anti-horário, no valor de β e em torno do eixo y. A matrix de rotação é dada por:

$$R_y(\beta) = \begin{pmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{pmatrix}.$$

3) O roll é uma rotação no sentido anti-horário, no valor de γ e em torno do eixo x. A matrix de rotação é dada por:

$$R_x(\gamma) = \begin{pmatrix} 1 & 0 & 0 \\ 0 & \cos \gamma & -\sin \gamma \\ 0 & \sin \gamma & \cos \gamma \end{pmatrix}.$$

Rotações em torno do yaw, pitch e roll podem posicionar um objeto 3D em qualquer orientação no espaço. Uma única rotação pode ser realizada através da multiplicação das matrizes yaw, pitch e roll. Realizando esta multiplicação, obtemos:

$$R(\alpha, \beta, \gamma) = R_z(\alpha) R_y(\beta) R_x(\gamma) = \begin{pmatrix} \cos \alpha \cos \beta & \cos \alpha \sin \beta \sin \gamma - \sin \alpha \cos \gamma & \cos \alpha \sin \beta \cos \gamma + \sin \alpha \sin \gamma \\ \sin \alpha \cos \beta & \sin \alpha \sin \beta \sin \gamma + \cos \alpha \cos \gamma & \sin \alpha \sin \beta \cos \gamma - \cos \alpha \sin \gamma \\ -\sin \beta & \cos \beta \sin \gamma & \cos \beta \cos \gamma \end{pmatrix}.$$

É importante notar que a ordem na qual as rotações yaw, pitch e roll acontecem, alteram o resultado da matriz $R(\alpha, \beta, \gamma)$.

2.4.2 Matriz Jacobiana, Velocidade e Torque

A matriz jacobiana tem sido uma ferramenta muito importante e altamente utilizada em aplicações de modelagem e controle de sistemas dinâmicos. Neste cenário, o Jacobiano

representa uma transformação matricial de posição para velocidade. Formalmente, o Jacobiano é um conjunto de derivadas parciais, onde:

$$x = \begin{pmatrix} x \\ y \\ z \\ \omega_x \\ \omega_y \\ \omega_z \end{pmatrix}$$

$$q = \begin{bmatrix} \alpha \\ \beta \\ \gamma \end{bmatrix}$$

$$J = \frac{\partial x}{\partial q}$$

Após realizar uma simples manipulação, também representamos a fórmula como:

$$\dot{x} = J \cdot \dot{q}$$

Onde o vetor \dot{x} representa a derivada da posição e orientação do último corpo rígido fixado à estrutura do sistema, e \dot{q} representa a derivada do vetor de coordenadas generalizadas, o que neste caso são os próprios ângulos yaw, pitch, roll.

Uma vez feito o cálculo da matriz Jacobiana, a mesma pode ser utilizada tanto para calcular a velocidade, quanto o torque do último corpo rígido fixado ao pitch. Considerando que o dispositivo será caracterizado como engastado em uma de suas extremidades, podemos fazer o cálculo do torque para manter o objeto em equilíbrio dinâmico, através da equação a seguir:

$$\vec{\tau} = J_{obj}^T \cdot \vec{F}$$

Onde: $\tau = [\tau_{roll} \tau_{pitch} \tau_{yaw}]^T$, $F = [F_x F_y F_z T_x T_y T_z]^T$ e $F_z = m_{obj} \cdot g$.

Observação: J_{obj} representa o jacobiano para o local do objeto.

2.5 SENSORES INERCIAIS

2.5.1 Acelerômetros

Sensores de aceleração, mais conhecidos como acelerômetros, são o carro chefe dos sensores inerciais, já que são os mais utilizados e conhecidos. O desenvolvimento desses dispositivos, normalmente, é acompanhado da preocupação em se conseguir componentes menores e mais baratos, com sensoriamento tridimensional e processamento digital de sinais embarcado no próprio chip.

Há uma infinidade de acelerômetros disponíveis no mercado e, atualmente, eles são empregados em grande escala em produtos de grande produção, como automóveis, consoles de videogame, smartphones, monitores de atividade física, etc. Nessas aplicações, sensores menores e mais baratos são preferidos em detrimento a acelerômetros de alta performance.

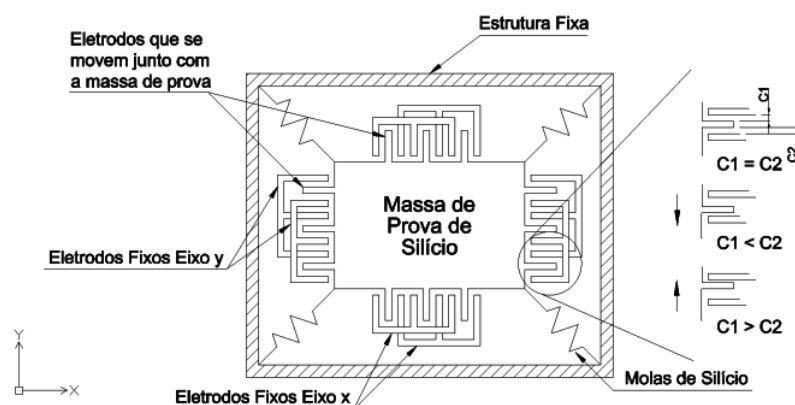


Figura 12: Estrutura de um acelerômetro de dois eixos

Fonte: < http://www.embarcados.com.br/wp-content/uploads/2015/01/aceler%C3%B4metro-de-dois-eixos_edited1-660x333.png > Acesso em: Jun. 2016

Basicamente, uma massa de prova de silício é suspensa e suportada por molas, também de silício, conferindo a esta massa uma mobilidade conhecida. O acelerômetro, por estar fixado ao sistema a ser medido, sofrerá a ação das mesmas forças impostas a este sistema. Uma variação de velocidade provocará, pelo princípio da inércia, uma alteração entre a distância dos eletrodos fixos e dos eletrodos móveis ligados à massa de prova. Em repouso, as capacitâncias $C1$ e $C2$ são iguais. Sob efeito da aceleração, $C1$ e $C2$ variarão dependendo do sentido e intensidade das forças que provoquem essa mudança de estado inercial. Daí em diante, a microeletrônica assume a tarefa de tratar os sinais elétricos provenientes destas variações de capacitância.

Em aplicações que requerem maior resolução e estabilidade, como detecção de terremotos, sistemas de navegação inercial e obtenção de perfis de reflexão sísmica, o mais indicado é o uso de acelerômetros baseados em estruturas vibrantes.

2.5.2 Giroscópios

Giroscópios tornaram possível que o homem pisasse na Lua. Com apenas isso em vista, já podemos ter uma ideia da importância desse componente para a humanidade. É graças a ele, também, que aviões conseguem sobrevoar em regiões de visibilidade zero com segurança, dentre inúmeras aplicações. Enfim, giroscópios são dispositivos fundamentais para orientação de espaçonaves.

Estruturalmente, um giroscópio é similar a um acelerômetro, possuindo, também, uma massa de prova suportada por molas de silício. A principal diferença é que a velocidade angular é obtida medindo-se a força Coriolis exercida na massa vibrante, ou seja, o movimento do corpo de prova deve possuir, pelo menos, dois graus de liberdade.

Observe a estrutura de um giroscópio baseado em MEMS, capaz de medir a velocidade angular em torno de somente um eixo:

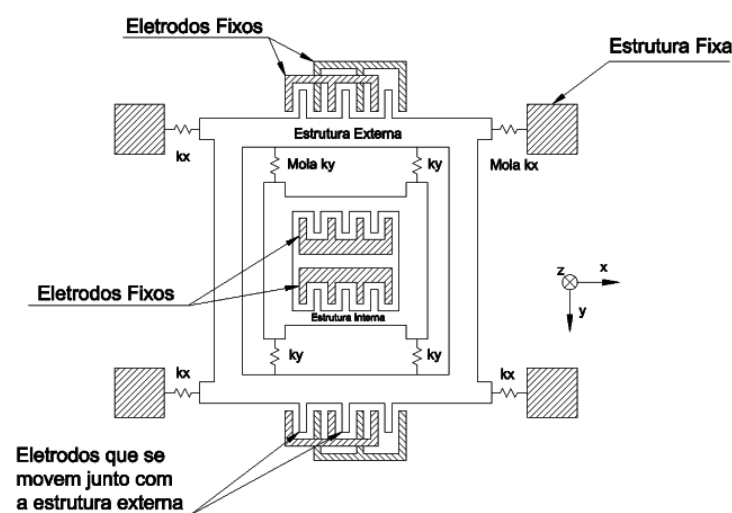


Figura 13: Estrutura de um giroscópio de um eixo

Fonte: < http://www.embarcados.com.br/wp-content/uploads/2015/01/Girosc%C3%B3pio-de-um-eixo_edited1-660x452.png> Acesso em: Jun. 2016

Há dois conjuntos de molas de silício: K_x , suportando a estrutura externa e conferindo mobilidade na direção do eixo x ; e K_y permitindo uma constante vibração da parte interna na direção do eixo y , causada por atuadores eletrostáticos. A força Coriolis proveniente de uma

rotação em torno do eixo z faz com que tanto a parte central quanto a parte externa da estrutura se movam na direção x. Este movimento da parte externa causa uma variação da capacitância entre eletrodos fixos e móveis conectados a esta parte da estrutura, mesmo princípio de funcionamento do acelerômetro explicado anteriormente. A grande vantagem do giroscópio se deve ao fato dele funcionar perfeitamente em ambientes de gravidade zero, o que permite seu uso em equipamentos desenvolvidos para exploração espacial.

2.5.3 IMU

Vários giroscópios e acelerômetros podem ser combinados em um mesmo dispositivo gerando, assim, sistemas inerciais completos. Um sistema composto por 3 giroscópios, capaz de medir a velocidade angular em três eixos ortogonais é conhecido como Unidade de Referência Inercial (IRU – Inertial Reference Unit), normalmente empregado em estabilização e direcionamento.

Adicionando a uma IRU a capacidade de se medir a aceleração em cada um dos três eixos através do uso de três acelerômetros, obteremos uma Unidade de Medida Inercial (IMU – Inertial Measurement Unit). Por poderem ser empregadas em sistemas de até 6 graus de liberdade (deslocamento linear e angular em cada um dos 3 eixos), IMU's são amplamente utilizadas para estabilização, determinação de posicionamento e controle de voo. Apesar de não ser tão trivial, utilizando o cálculo integral podemos determinar a velocidade de um objeto a partir da aceleração e sua posição angular a partir da velocidade angular.

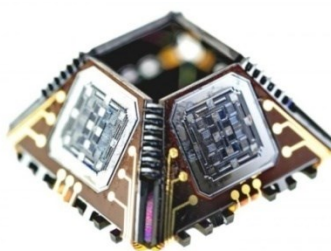


Figura 14: IMU multiaxial baseado na abordagem "folded MEMS"

Fonte: < http://www.embarcados.com.br/wp-content/uploads/2015/01/imu_multiaxial-660x440.jpg > Acesso em Jun. 2016

Finalmente, agregando a uma IMU um computador de navegação para processamento dos sinais de cada um dos 6 sensores, criamos um Sistema de Navegação Inercial (INS – Inertial Navigation System). Neste caso, não trata-se somente de um circuito integrado, mas sim de um sistema embarcado completo, capaz de realizar a fusão sensorial dos sensores

inerciais através de algoritmos que agregam os dados de cada um dos sensores individuais e identificam um padrão de movimento, proporcionando dados de velocidade e posicionamento para controle de atitude e direção.

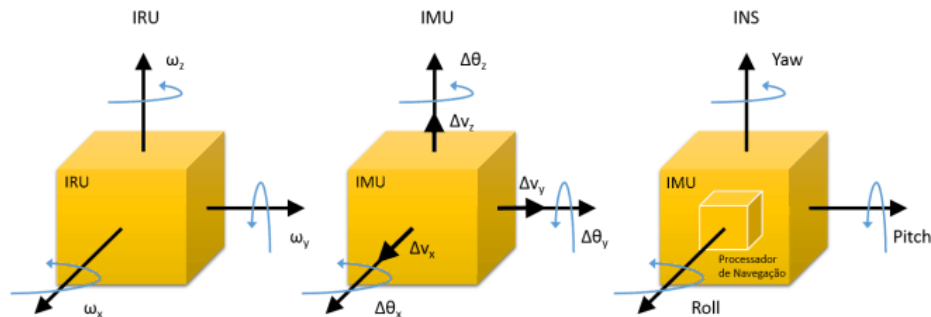


Figura 15: Unidades inerciais

Fonte: < <http://www.embarcados.com.br/wp-content/uploads/2015/01/IRU-IMU-INS-660x229.png> > Acesso em: Jun. 2016

2.5.3.1 IMU MPU-6050

O sensor utilizado neste projeto é o MPU-6050, que combina 3 eixos com giroscópio e 3 eixos com acelerômetros. A comunicação é feita por meio do protocolo I2C.

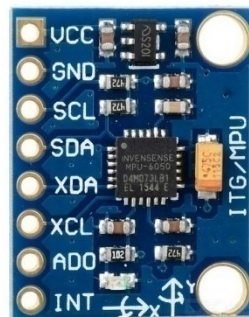


Figura 16: MPU-6050

Fonte: < http://img.dxcn.com/productimages/sku_154602_2.jpg > Acesso em: Jun. 2016

Especificações:

Modelo: GY-521

Cor: Azul

Material: PCB, Plástico e Cobre

Chip: MPU-6050

Bateria: 3~5V

Modo de Comunicação: padrão protocolo I2C

Faixa de uso do Giroscópio: +/- 250 500 1000 2000 degree/sec

Faixa de uso da Aceleração: +/- 2g, +/- 4g, +/- 8g, +/- 16g

Dimensões: 2.1 cm x 1.6 cm x 0.3 cm

2.5.3.2 Cálculo da orientação a partir de dados do acelerômetro

Para realizar o cálculo da orientação, o acelerômetro depende de uma força gravitacional constante, com massa no valor de 1g. Caso não existam forças adicionais no sistema, a magnitude da aceleração é 1g, e a rotação dos sensores podem ser calculados utilizando o vetor posições do vetor de aceleração.

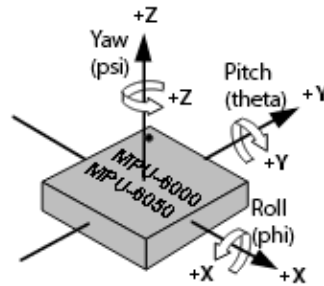


Figura 17: Eixos IMU MPU-6050

Fonte: < http://www.embarcados.com.br/wp-content/uploads/2015/01/MPU6050_820x609.png> Acesso em Jun. 2016

Nota: Se o eixo z estiver alinhado com o vetor de força gravitacional, será impossível calcular a rotação em torno do eixo z utilizando o acelerômetro.

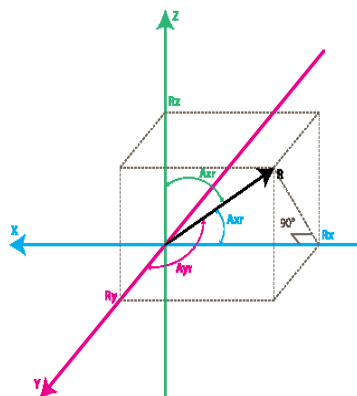


Figura 18: Forças acelerômetro

Fonte: < http://www.starlino.com/imu_guide.html> Acesso em: Jun. 2016

$$\cos(A_{xr}) = \frac{R_x}{R}$$

$$\cos(A_{yr}) = \frac{R_y}{R}$$

$$\cos(A_{zr}) = \frac{R_z}{R}$$

$$R = \sqrt{R_x^2 + R_y^2 + R_z^2}$$

$$A_{xr} = \arccos\left(\frac{R_x}{R}\right)$$

$$A_{yr} = \arccos\left(\frac{R_y}{R}\right)$$

$$A_{zr} = \arccos\left(\frac{R_z}{R}\right)$$

2.5.3.3 Cálculo da orientação a partir de dados do giroscópio

O giroscópio mede a taxa de variação de um ângulo. Também é possível medir a orientação à partir do giroscópio, para isto é necessário iniciar o sensor em uma posição conhecida, e medir a velocidade angular em torno dos eixos X, Y e Z em intervalos de tempo definidos.

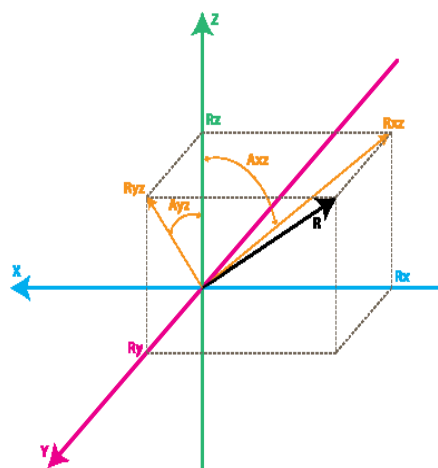


Figura 19: Forças no giroscópio

Fonte: < http://www.starlino.com/imu_guide.html > Acesso em Jun. 2016

Inicialmente colhemos uma medida inicial vindo do acelerômetro:

$$\tan(A_{xz}) = \frac{R_x}{R_z}$$

$$A_{xz} = \text{atan2}(R_x, R_z)$$

Após a coleta do primeiro dado, podemos utilizar o giroscópio:

$$A_{xz}(n-1) = \text{atan2}(R_{x\text{Est}}(n-1), R_{z\text{Est}}(n-1))$$

$$A_{xz}(n) = A_{xz}(n-1) + \text{Taxa}A_{xz}(n) * t$$

Onde:

$\text{Taxa}A_{xz}$: Taxa de variação entre o valor de ângulo atual e o anterior.

2.5.3.4 Algoritmos de Fusão

Com o intuito de melhorar a precisão da estimação dos valores de rotação, dados do acelerômetro são utilizados juntamente com dados do giroscópio através de um algoritmo de fusão, melhorando assim a qualidade da resposta do sensor.

$$R_{\text{est}}(n) = \frac{R_{\text{acc}} \cdot w_1 + R_{\text{gyro}} \cdot w_2}{w_1 + w_2}$$

Onde w_1 e w_2 são constantes e a soma das mesmas, deve ser diferente de zero.

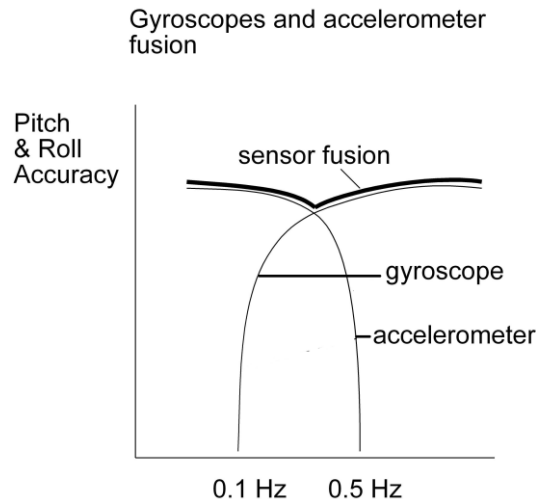


Figura 20: Fusão de Sensores

Fonte: <<http://memscentral.com/Images-case-notes-perspec/sensor-fusion-low-res.jpg>> Acesso em: Jun. 2016

2.6 MOTORES COM E SEM ESCOVAS

Os motores de corrente elétrica contínua atuais diferem na forma como a corrente é transferida para o comutador ou eletroímãs que fazem com que o rotor continue a rodar. Essencialmente, em um motor escovado a corrente é transferida mecanicamente através de escovas metálicas, enquanto que em um motor sem escovas, o rotor é ligado eletronicamente, sem a necessidade de contatos físicos.

2.6.1 Motor escovado DC

Motores elétricos de corrente contínua funcionam através da criação de campos magnéticos, cuja atração e repulsão mantêm o giro do rotor central. Num motor escovado, ímãs fixos são colocadas em ambos os lados de um eletroímã rotatório, um orientado para o pólo positivo e o outro para o negativo. O eletroímã é formado por uma série de bobinas (normalmente três, colocadas em pontos equidistantes em torno do rotor), chamado comutador. Quando a eletricidade passa, essas bobinas geram o seu próprio campo magnético que é repelido e atraído pelo campo magnético gerado pelos ímãs fixos. A corrente é transferida para as bobinas do comutador por escovas metálicas que rodam juntamente com o rotor. Quando o motor é ligado, a corrente passa para os eletroímãs cujos campos magnéticos são repelidos por um ímã fixo e atraído pelo outro, fazendo com que o rotor gire. Com esse giro, as escovas metálicas entram e saem em contato com cada bobina em série, assim a oposição e a atração entre os campos magnéticos resultantes e os campos dos ímãs estáticos mantêm o giro do eletroímã.

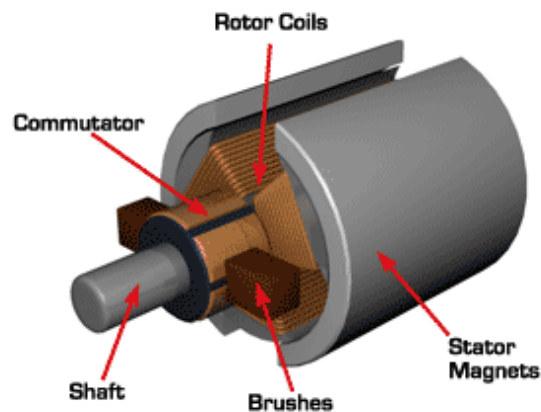


Figura 21: Motor DC com escovas

Fonte: < <http://www.electrical4u.com/images/PMDC-motor-construction.gif> > Acesso em: Jun. 2016

A principal vantagem dos motores DC com escovas é que são mais baratos e mais simples de construir e manter do que os motores sem escovas, pois o mecanismo é menos complicado.

2.6.2 Motor sem escovas (brushless)

Em um motor de corrente contínua sem escovas, as posições dos ímãs fixos e das bobinas eletromagnetizadas são invertidas. Os ímãs fixos são agora colocados sobre o rotor e as bobinas são colocadas em torno do invólucro. O motor funciona através da corrente que é passada através de cada bobina em série, de modo a repelir e atrair os campos dos ímãs fixos e manter o rotor anexado a ele em sua volta. Para um motor desse tipo funcionar, as bobinas do comutador devem ser mantidas sincronizadas com os ímãs fixos para que os campos estejam continuamente em oposição e o rotor seja mantido em rotação. Isso requer um controlador eletrônico ou um microprocessador para coordenar a aplicação da corrente para cada bobina eletromagnética.



Figura 22: Motor brushless

Fonte: Autor "adaptado de" FPV flying

A principal vantagem dos motores sem escovas é que a transferência da corrente para o comutador não é mecânica. Como os motores escovados dependem do contato físico das escovas de metal com as bobinas do comutador, estão sujeitos a uma perda de eficiência devido ao atrito com os contatos e também há um desgaste de todas as peças mecânicas nas escovas e nas conexões depois de longos períodos de utilização, devido ao atrito. Como motores sem escovas esquentam menos (devido à falta de atrito) podem trabalhar em velocidades maiores (pois o aquecimento interfere nos campos magnéticos).

2.6.3 Motor brushless GBM2208H-80T

O motor utilizado neste projeto é o GBM2208H-80T. Este motor é projetado para plataformas multi-rotor de grande porte que procuram levantar câmeras do tamanho gopro - 0,8 kg/cm. A estabilização da câmera é feita usando motores diretos sem escovas, gimbals baseados em motores BLDC são muito semelhantes aos gimbals regulares com base em servo.



Figura 23: Motor GBM2208H-80T

Fonte: < <http://www.iflight-rc.com/ipower-gbm2208h-80t-gimbal-motor.html>> Acesso em: Jun. 2016

Especificações:

Peso: 43g

Dimensões do motor: 28x22mm

Dimensões do estator :22x8mm

Diâmetro do fio de cobre: 0.16mm

Configuração: 12N14P

Resistência: 14.6 ohms

Método de terminação: estilo estrela

Pré-ferido com 80 voltas e 3.17mm de eixo

Furos de montagem do fundo: 16 e 19mm de centro a centro

Furos de montagem do topo: 12mm de centro a centro

Alcance da câmera: 100-200g (tamanho GoPro)

2.6.4 Atuador dinâmico

Os torques do motor são produzidos por três atuadores, três motores DC. Uma vez que é possível controlar a voltagem do motor DC ao invés do torque diretamente, é necessário encontrar a relação entre a tensão e o torque. Um motor DC funciona basicamente pelo princípio de que um condutor de corrente em um campo magnético sofre uma força. A magnitude do torque é dada pela equação abaixo:

$$\tau = K_m * i$$

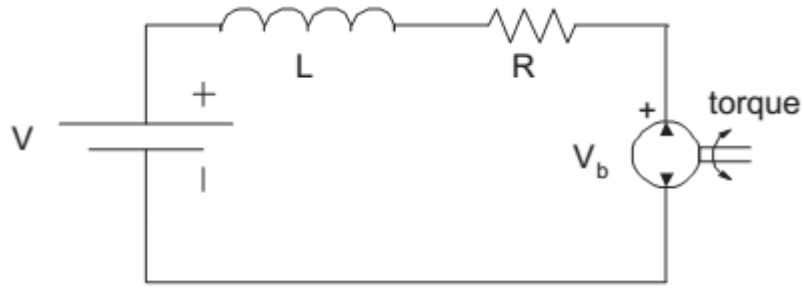


Figura 24: Modelo dinâmico do atuador

Fonte: <<http://www.hindawi.com/journals/mpe/aa/879581/>> Acesso em Nov. 2015

Onde K_m é uma constante física do torque e i é a corrente. Além disso, um condutor movido em um campo magnético irá gerar uma Força Eletromotriz proporcional a velocidade e no sentido contrário do fluxo da corrente. Assim, a Força Eletromotriz V_b é:

$$V_b = K_b * w$$

Onde K_b é constante física da velocidade e w é a velocidade angular.

Considerando o circuito da Figura 7 para uma armadura controlada de motor DC, é possível expressar a corrente com a seguinte equação diferencial:

$$L \frac{d_i}{d_t} + R_i = V - V_b$$

Entretanto, ignora-se a indutância L e depois o torque do atuador pode ser expresso como:

$$\tau = \frac{K_m}{R} * (V - K_b * w)$$

Assim, é possível expressar os torques do atuador no sistema autoestabilizador pela equação abaixo:

$$\tau_i = \frac{K_{m_i}}{R_i} * (V_i - K_{b_i} * w_i) \text{ onde } i = 1, 2, 3$$

2.7 SPWM

A maioria dos motores brushless que vêm sendo utilizados, são para mecanismos de alta velocidade de operação, tais como drones, robôs móveis, entre outros. Para este propósito, é necessário saber a posição do rotor, portanto mede-se a Força Eletromotriz Reversa ou utiliza-se um sensor Hall. Porém para aplicações de Controle de Posição, observa-se que o SPWM é eficaz.

Deste modo o PWM utiliza sinais de modulação do tipo senoidal, pré-definidos para determinar tensões de saída. Abaixo segue um exemplo de um teste realizado com o componente L6234.

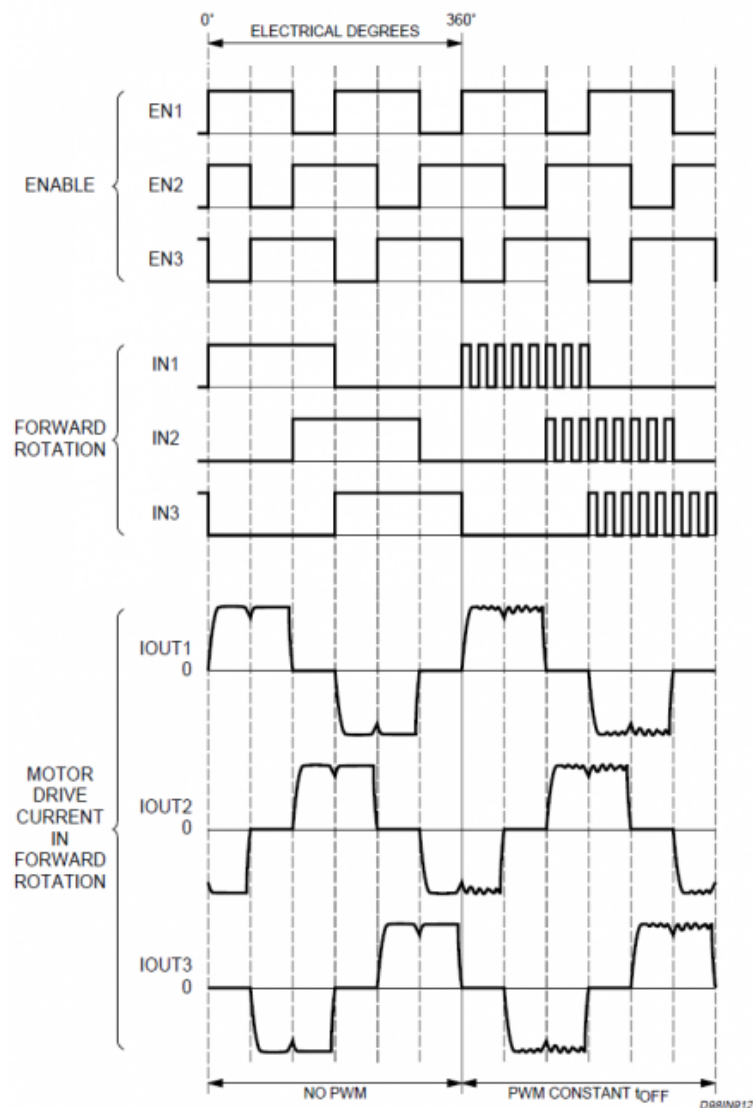


Figura 25: Componente L6234 operando em altas velocidades

Fonte: <<http://www.berryjam.eu/2015/04/driving-bldc-gimbals-at-super-slow-speeds-with-arduino/>> Acesso em Jun. 2016

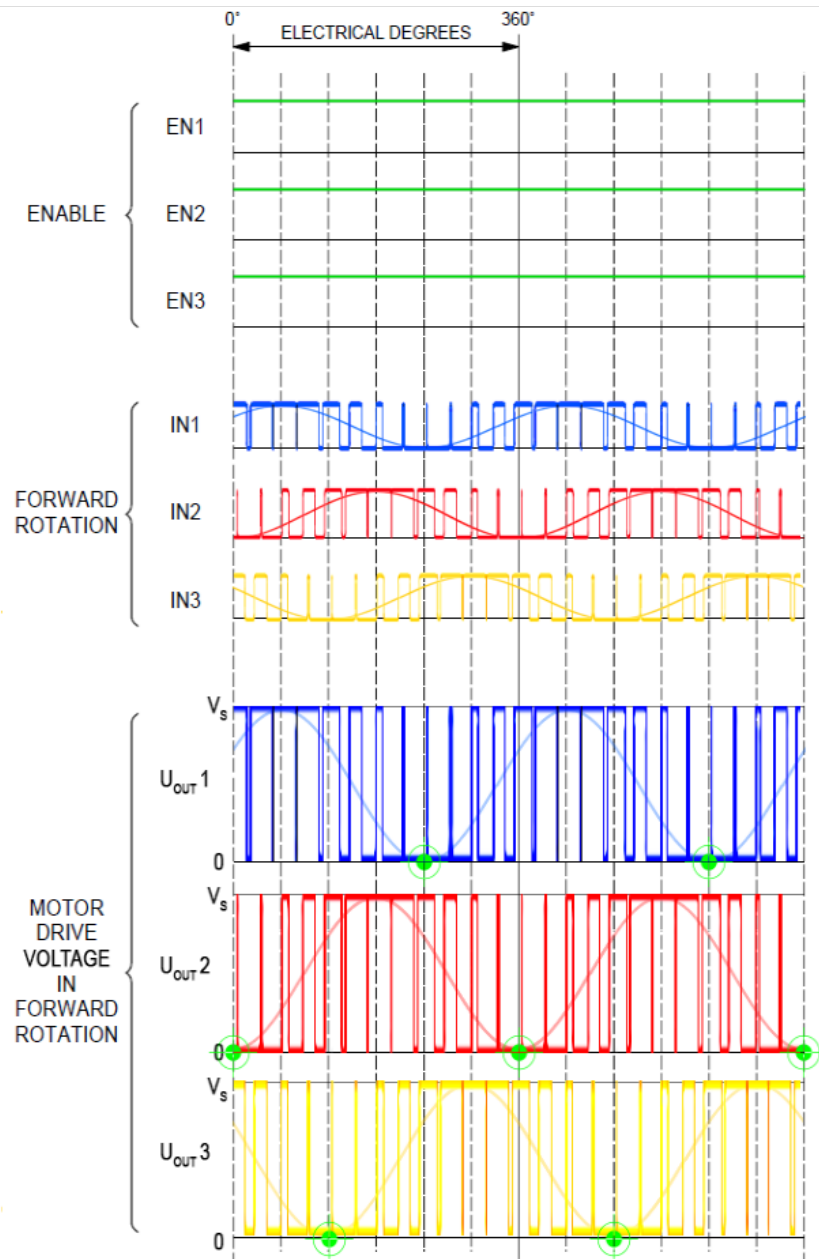


Figura 26: L6234 operando em baixa velocidade (Controle de Posição)

Fonte: <<http://www.berryjam.eu/2015/04/driving-bldc-gimbals-at-super-slow-speeds-with-arduino/>> Acesso em Jun. 2016

2.8 DIAGRAMA DE BLOCOS

Existe um botão onde é possível o usuário indicar a orientação inicial desejada. Este é a referência e entrada do sistema de controle. A diferença entre a orientação atual e a orientação indicada pelo IMU resulta no erro que será corrigido através da programação realizada no microcontrolador. Este calcula os esforços necessários em cada motor para corrigir a posição e envia três pulsos PWM para cada um. Os três motores são acionados com o intuito de corrigir o erro calculado pelo microcontrolador. A leitura da orientação feita pelo IMU, o cálculo do erro feito pelo microcontrolador e o envio do sinal PWM ocorrem em um período muito curto, de forma que o erro não fica aparente. Abaixo segue o esquema do diagrama de blocos:

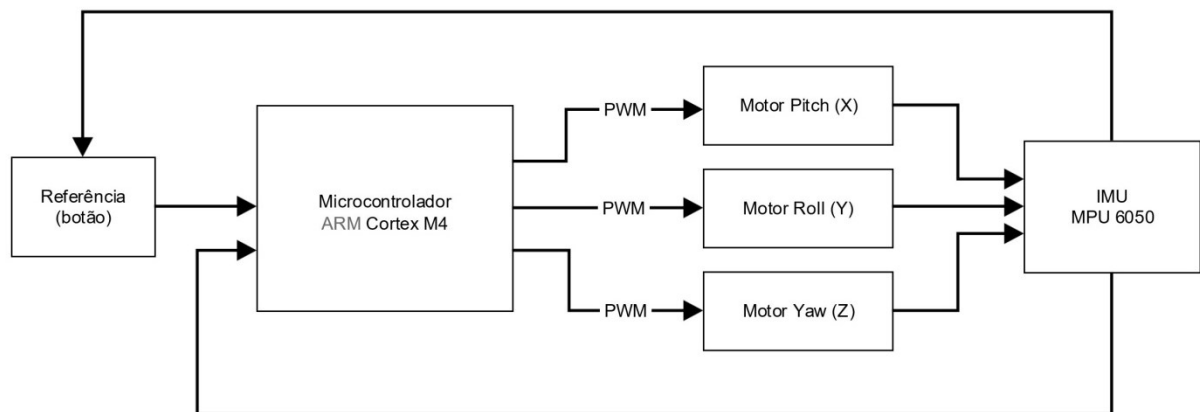


Figura 27: Diagrama de Blocos

Fonte: Autor

2.9 PRIMEIRA EXPERIÊNCIA

Através de um microcontrolador (Arm Cortex M4) foi desenvolvido e programado um código em linguagem C embarcado, capaz de enviar um sinal de PWM em forma senoidal. Com a montagem abaixo, alimentada por uma fonte variável, testamos o funcionamento do circuito integrado L6234, que contém internamente 3 meia-pontes H que, por meio do código enviado, habilita e coordena as três fases do motor.

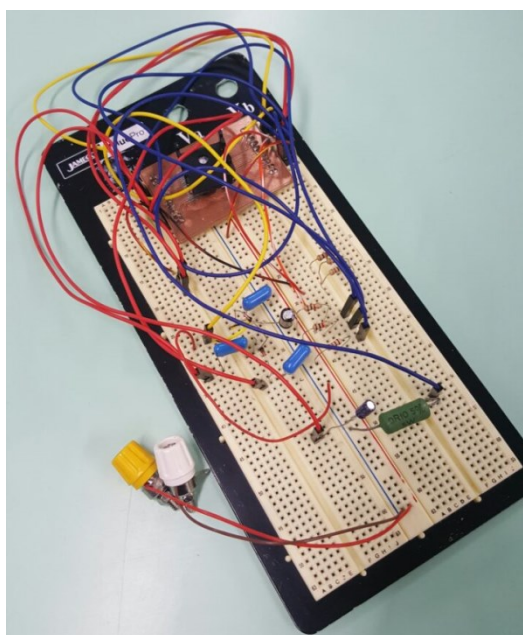


Figura 28: Circuito montado em Protoboard

Fonte: Autor

Em seguida, com o código aperfeiçoado fizemos o teste em conjunto com o sensor MPU 6050, que é composto por três acelerômetros e três giroscópios. Nesta etapa identificamos no sensor os seus eixos Pitch (X), Roll (Y) e Yaw (Z). Para isso, conectamos um motor e enviamos dados de apenas uma das coordenadas do sensor e manualmente rotacionamos o sensor até perceber a inversão de fase do motor. Esta etapa é muito importante para identificar a posição correta em que o sensor deve ser fixado na estrutura do dispositivo.

O último teste efetuado na protoboard foi a conexão de todos os motores simultaneamente e, agora, com a alimentação da bateria. Este teste foi realizado para aumentar as garantias de que as coordenadas obtidas na etapa anterior estavam corretas.

2.9 PLACA MICROPROCESSADORA GIMBAL

Para controlar os motores, um controlador especial foi concebido. Ele recebe informações de giroscópios e acelerômetros montados na plataforma de câmera. Um algoritmo padrão IMU é utilizado para definir os ângulos de inclinação da câmera. Através de um software desenvolvido, o operador seleciona o ângulo de inclinação desejado. O controlador PID calcula a quantidade de compensação e envia os comandos para a unidade de

potência, que controla a corrente no enrolamento e, assim, a direção do vetor de campo magnético no estator. O campo magnético move o rotor para a posição correta.

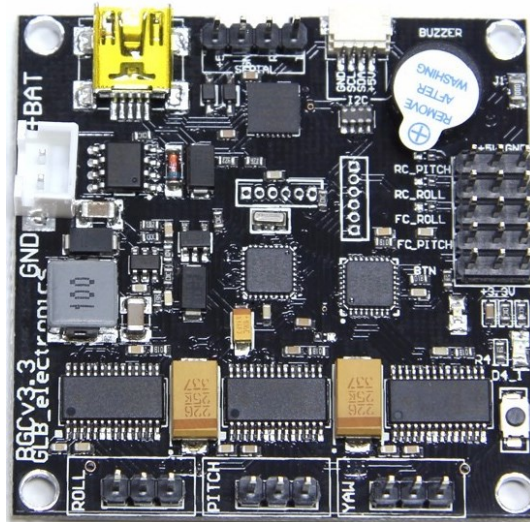


Figura 29: Placa BaseCam

Fonte: <http://mlb-s1-p.mlstatic.com/placa-controladora-3-eixos-gimbal-alexmos-bgc-drv8313-284911-MLB20658673254_042016-F.jpg> Acesso em Jun. 2016

Especificações:

Número de eixos estabilizados: 3

Potência: 8..18V (3s-4s LiPo)

Máxima corrente de cada motor DC: 2.8A

Limite de corrente (curto) de cada motor: 7.1A

Frequência de saída: 8 or 32 KHz PWM

Sensor de conexão: I2C

Sensores inerciais: MPU6050

Comunicação com PC: on-board USB

Proteção de frequência: Sim

Proteção de superaquecimento: Sim

Proteção de tensão: Sim

Proteção de polaridade reversa: Sim

Dimensões: 50mm x 50mm

Furos de montagem: M3, 45mm

2.9.1 Microcontrolador

O Microcontrolador utilizado neste projeto é o ARM Cortex-M4. ARM Cortex-M é uma família de núcleos de processador RISC de 32-bit licenciados pela ARM Holdings, principalmente usados em microcontroladores. Os núcleos que compõem a família são Cortex-M0, Cortex-M1, Cortex-M3, Cortex-M4.

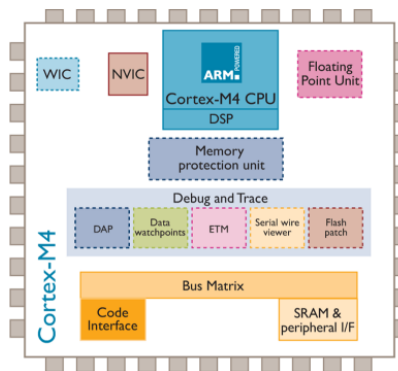


Figura 30: Arquitetura ARM Cortex M4

Fonte: <<http://www.mouser.com/images/microsites/cortexM4.png>> Acesso em Jun. 2016

Conceitualmente, o Cortex-M4 é um Cortex-M3 com Instruções DSP, e uma Unidade de Ponto Flutuante opcional. Se o núcleo contém a unidade de ponto flutuante, é conhecido como Cortex-M4F. As principais características do núcleo Cortex-M4 são:

Arquitetura ARMv7-ME;

Suporte a instruções:

-Thumb (total)

-Thumb-2 (total)

-Multiplicador de hardware de 1 ciclo (32-bit), divisão de hardware de 2-12 ciclos (32-bit)

-Extensão DSP: 16/32-bit MAC de único ciclo, 16-bit MAC duplo de único ciclo, aritmética 8/16-bit SIMD.

-Extensão de ponto flutuante (silicon option): Unidade de ponto flutuante de precisão simples, chamada FPv4-SP.

Pipeline de 3 estágios com branch predictor;

1 a 240 interrupções de hardware, mais NMI;

Latência de interrupções de 12 ciclos;

Modos sleep integrados;

MPU com 8 regiões (silicon option);

1,25 DMIPS/MHz

2.9.2 Driver de Motor Trifásico

Neste projeto foi utilizado o driver L6234, o qual é apropriado para este tipo de equipamento por ser uma meia ponte tripla que serve para conduzir um motor brushless DC.

A tecnologia utilizada BCD multipower combina isoladas transistores de potência DMOS com circuitos CMOS e bipolares no mesmo chip.

Ao utilizar a tecnologia mista foi possível otimizar o circuito lógico e o estágio pôde alcançar o melhor desempenho possível.

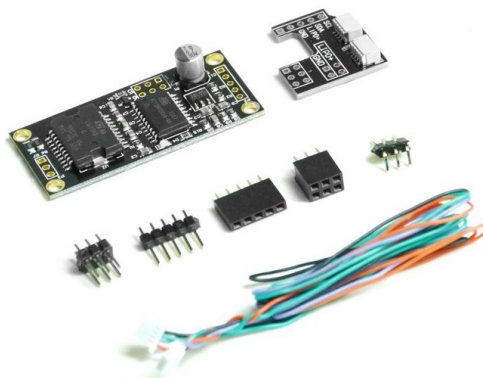


Figura 31: Kit L6234

Fonte: <<https://www.basecamelectronics.com/simplebgc/>> Acesso em Jun. 2016

Especificações:

Tipo: Driver

Tensão de alimentação operacional: 7 V to 52 V

Corrente de Alimentação Operativa: 7 mA

Estilo de montagem: Through Hole

Caixa / Gabinete: PDIP-20

Embalagem: Tube

Marca: STMicroelectronics

Faixa de temperatura operacional: - 40 C to + 125 C

Série: L6234

Peso unitário: 2.260g

2.9.3 Conexão da placa microprocessadora

Abaixo segue um esquemático da montagem da placa juntamente com IMU e motores:

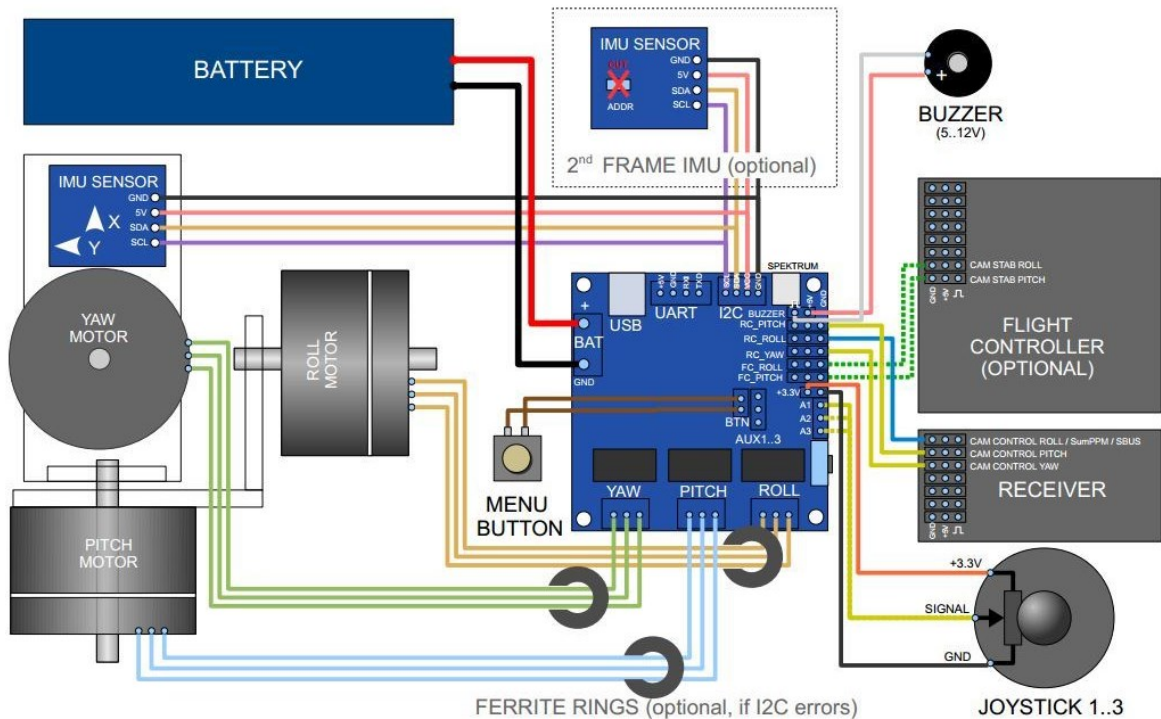


Figura 32: Montagem placa didática

Fonte: <<http://www.dronetrest.com/uploads/db5290/optimized/2X/7.JPG>> Acesso em: Jun. 2016

Em termos de montagem do IMU e da câmera, é possível se referir pela imagem abaixo. O IMU é o que tem duas conexões de 4 pinos sobre ele. As montagens da câmera sobre a placa de montagem e o IMU precisam ser montados por cima do motor do eixo yaw.

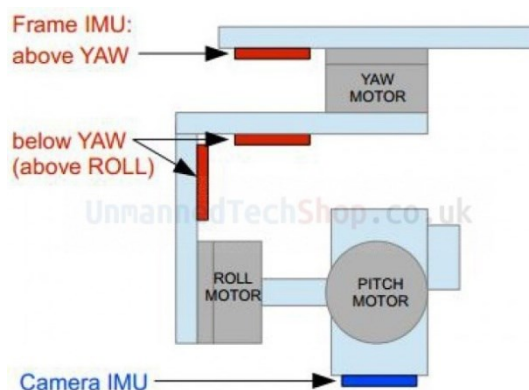


Figura 33: Localização IMU

Fonte: <<http://www.dronetrest.com/uploads/db5290/original/2X/3.JPG>> Acesso em: Jun. 2016

Para fazer a conexão com o computador, basta conectar o cabo USB e executar o software SimpleBGC_GUI. Neste é possível realizar calibrações do acelerômetro e giroscópio, sintonizar o PID, acompanhar os gráficos de erro, entre outras configurações.

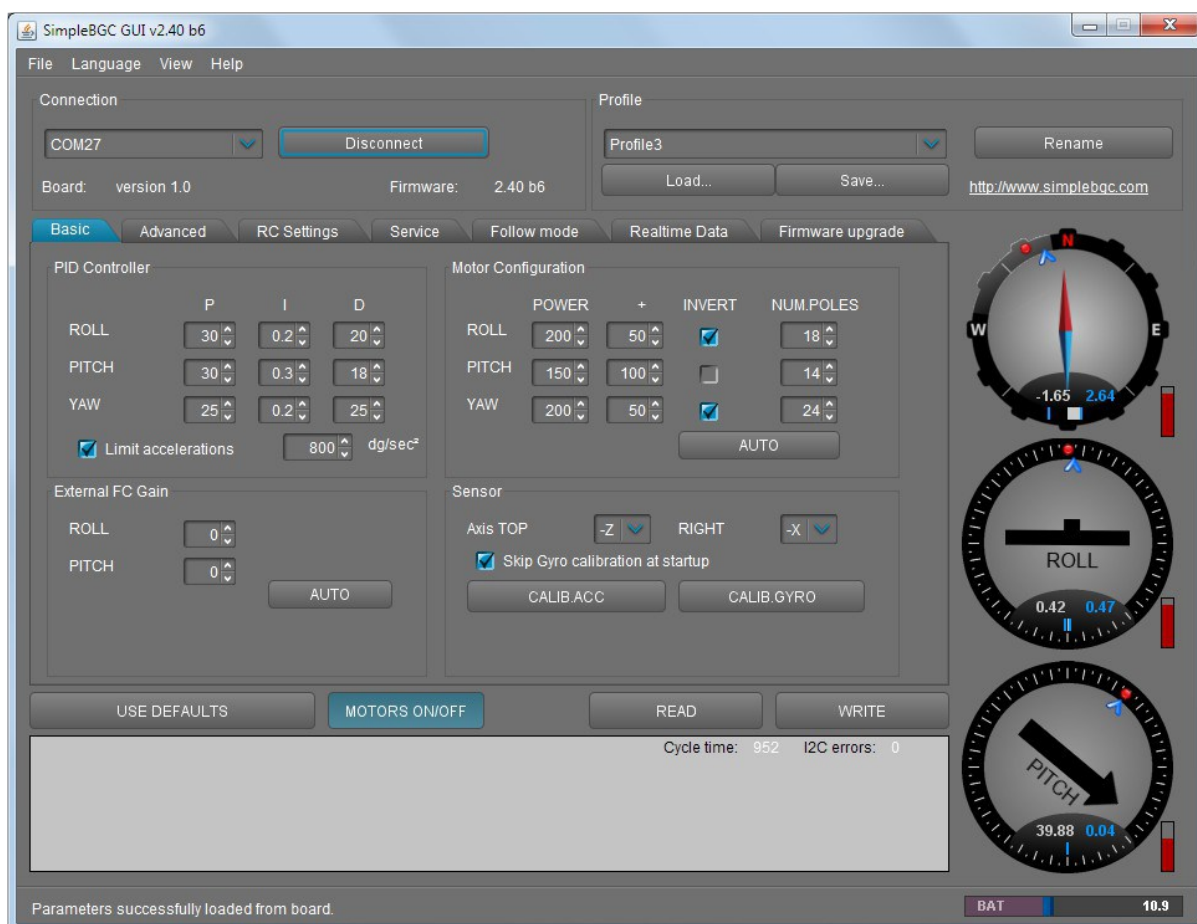


Figura 34: Tela de configurações

Fonte: SimpleBGC_manual_2_4_eng.pdf

É muito importante configurar corretamente a posição inicial dos pólos magnéticos do motor, porque todos os outros cálculos dependem desta informação. Para o eixo de rotação, ele permite ajuste fino da câmera em relação a montagem. Para os eixos pitch e roll, há uma opção para calibrar automaticamente.

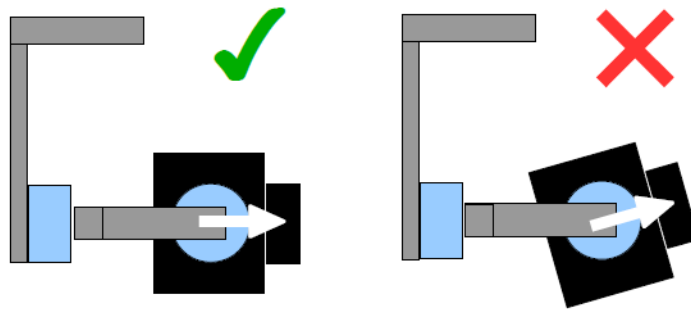


Figura 35: SetPoint

Fonte: SimpleBGC_manual_2_4_eng.pdf

2.9.4 Sintonização do PID

O ajuste, calibração ou sintonia do controlador, por métodos práticos, consiste em deduzir valores que vão permitir o cálculo dos parâmetros, partindo da resposta do sistema quando este é sujeito a entradas específicas. A vantagem deste método é não existir necessidade de conhecer o modelo do sistema, por muitas vezes difícil de determinar.

Previamente é importante apontar que para o controle dos 3 (três) eixos e recomendado que o dispositivo fique estável próximo da posição em que a câmera deverá permanecer quando estabilizada. Esse preparo garante um melhor resultado no controle ao reduzir o esforço necessário do motor.

Este controlador funciona com a composição de 3 (três) ganhos que possuem influências distintas sobre a resposta do controlador: o ganho proporcional (K_p), integrativo (K_i) e derivativo (K_d). Esses ganhos se sobrepõem conforme demonstrado pela imagem abaixo:

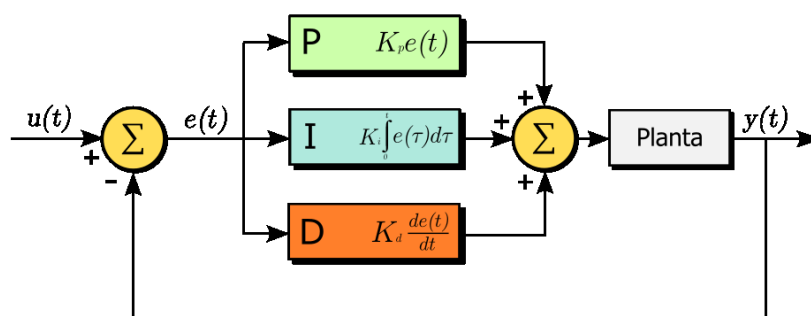


Figura 36: Controle PID

Fonte: <<https://upload.wikimedia.org/wikipedia/commons/thumb/2/2d/PID.svg/2000px-PID.svg.png>> Acesso em: Jun. 2016

A princípio foi definido que nenhum dos ganhos do controlador seriam iguais a zero, pois para a estabilização de uma imagem é muito importante ter um erro estacionário muito baixo, com correções rápidas, porém com o mínimo de oscilação em altas frequências, o que seria praticamente impossível de se conseguir sem os ganhos integrativos e derivativos, considerando a estrutura do *laxis*, que pode se tornar bem instável.

Definido isso começamos a sintonização estipulando ganhos unitários. Em seguida, para cada motor, partimos do zero e aumentamos gradativamente a potência fornecida até notar uma trepidação, e então é adotada uma potência logo abaixo. Vale ressaltar também que não é interessante adotar valores muito altos para a potência, pois ocasiona o aumento do consumo de energia e pode sobreaquecer os motores. Destacando que a escala dos ganhos varia de 0 a 255 os parâmetros iniciais ficaram:

Motores	K_P	K_I	K_D	Potência	Potência de trepidação*
Eixo X_{Pitch}	1	1	1	100	120
Eixo Y_{Roll}	1	1	1	100	120
Eixo Z_{Yaw}	1	1	1	130	150

Tabela 4: Parâmetros iniciais do PID

Fonte: Autor

*Potência em que a trepidação ficou aparente.

Para regular o ganho K_I é necessário forçar uma variação na planta e analisar o tempo de correção do controlador. Quanto maior este ganho, mais rápida será a correção, porém ganhos exagerados levam o sistema a instabilidade. Pela dificuldade em manter o sistema estável a influência do ganho integrativo foi fortemente reduzida.

Em seguida são incrementados os ganhos proporcionais (K_P) até aparecerem trepidações novamente e escolhido o maior ganho que manteve o sistema estável.

O ganho K_D tem como principal característica o amortecimento da resposta da planta. Quanto mais elevado, mais suave será a transição, reduzindo assim trepidações. Contudo ganhos demasiados reduzem o tempo de resposta do sistema, podendo deixar a correção muito lenta.

Após uma primeira análise dos ganhos os parâmetros se alteraram para:

Motores	K_P	K_I	K_D	Potência
Eixo X_{Pitch}	14	0,08	10	100
Eixo Y_{Roll}	12	0,05	20	100
Eixo Z_{Yaw}	15	0,1	10	130

Tabela 5: Segunda tentativa de parâmetros

Fonte: Autor

Os testes são repetidos até que a resposta da planta esteja adequada. Ao final consideramos rever também os valores adotados para a potência, de tal forma que os parâmetros finais ficaram:

Motores	K_P	K_I	K_D	Potência
Eixo X_{Pitch}	14	0,08	30	120
Eixo Y_{Roll}	20	0,05	25	100
Eixo Z_{Yaw}	15	0,03	30	130

Tabela 6: Parâmetros finais

Fonte: Autor

Os ganhos calculados variam caso qualquer característica da planta mude, logo para controlador câmeras com pesos, centros de massa ou dimensões diferentes das utilizadas durante a sintonização do PID, será necessário um novo cálculo destes parâmetros. Porém os parâmetros gerados acima servem como referência para a novas sintonizações, já que são muito mais próximas das condições ideais do que ganhos unitários.

Apesar do objetivo ser a estabilização de câmeras, para demonstrar a total capacidade do *laxis*, foi impresso na impressora 3D uma base de aproximadamente 10 cm de diâmetro e feita a sintonização do PID para o equilíbrio de objetos distintos apoiados sobre essa placa.

A tabela abaixo apresenta os parâmetros do PID utilizados para controlar uma garrafa pet de 500 ml semi cheia apoiada sobre a base impressa:

Motores	K_P	K_I	K_D	Potência
Eixo X_{Pitch}	12	0,08	20	50
Eixo Y_{Roll}	14	0,06	25	100
Eixo Z_{Yaw}	20	0,1	30	100

Tabela 7: Parâmetros garrafa pet

Fonte: Autor

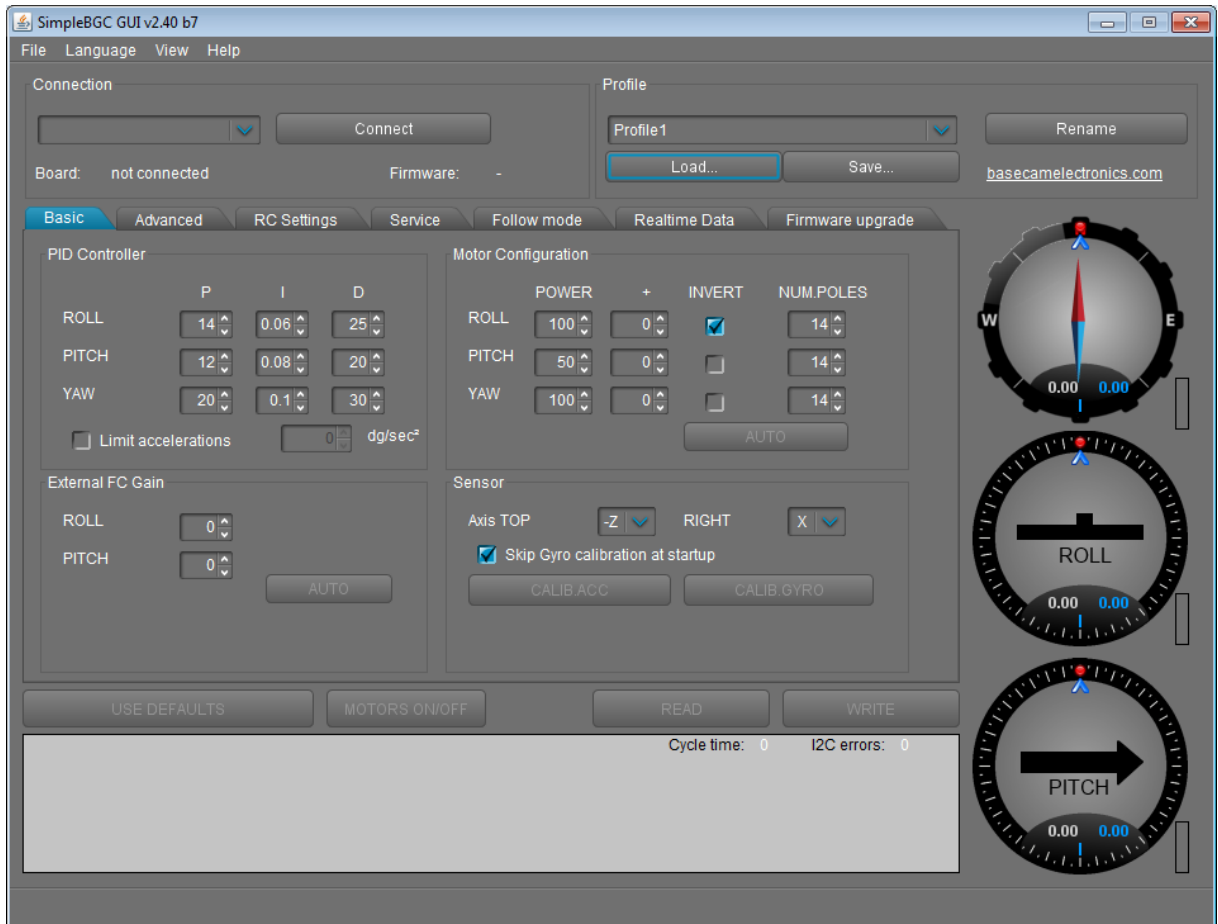


Figura 37: Parâmetros garrafa pet

Fonte: Autor

2.10 ESTRUTURA

2.10.1 Determinação da estrutura e posicionamento dos eixos

O primeiro conceito desenvolvido neste projeto foi o fato de que através da análise das equações cinemáticas e da estrutura mecânica do dispositivo, chegou-se à conclusão de que os três eixos ortogonais deveriam conter um ponto em comum entre si (Fig. 38). Isso garante com que a movimentação dos eixos seja independente, ou seja, uma movimentação no yaw, não afeta os ângulos roll e pitch. Também se observou que o torque disponível nos três motores, assim como a posição e orientação da última peça fixada ao yaw, são suficientes para a realização da estratégia de Controle PID, com realimentação da malha dada por um sensor do tipo IMU (Inertial Measurement Unit).

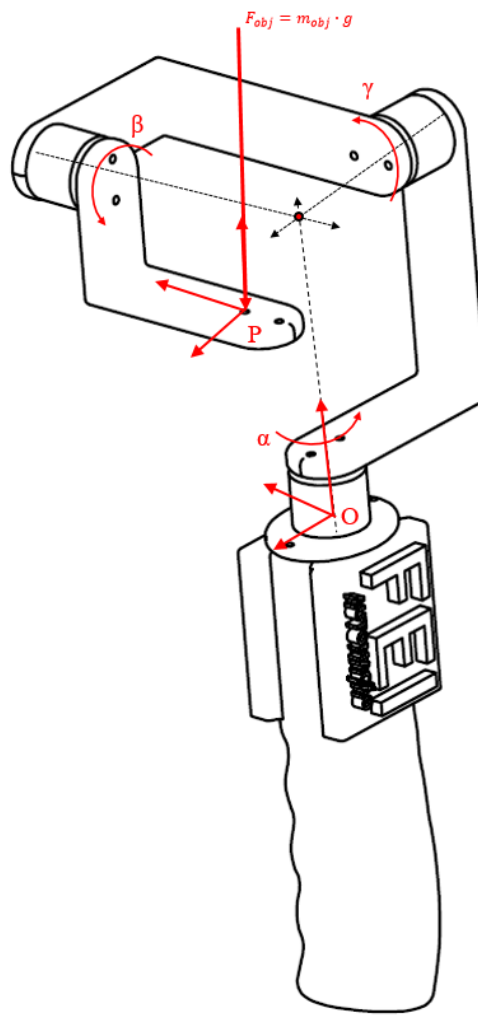


Figura 38: Posição das forças na estrutura

Fonte: Autor

2.10.2 Impressora 3D

Neste projeto foi utilizada a impressora 3D da fabricante Makerbot. Ideal para imprimir protótipos em ABS, a impressora 3d Makerbot 2X foi projetada para especialistas em impressão 3D. Trata-se de um equipamento da quarta geração da MakerBot, equipado com dois extrusores. Sua tecnologia traz duas cores entrelaçadas e impressas por meio de bicos distintos precisamente alinhados. E o melhor, sem pausar ou necessitar troca de filamentos.

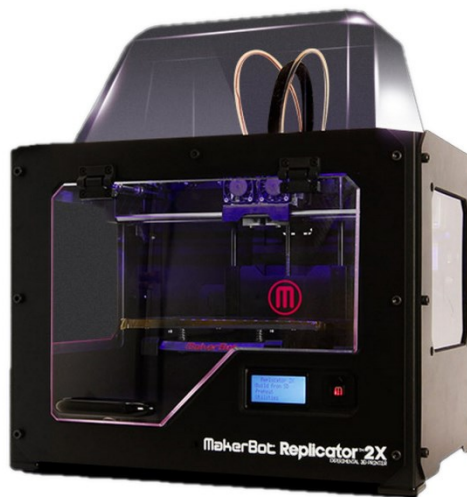


Figura 39: Impressora Makerbot 2X

Fonte: <<https://3dhubs.s3-eu-west-1.amazonaws.com/s3fs-public/Makerbot%20Replicator%202x.png>> Acesso em: Jun. 2016

Especificações:

Área de impressão: 25 x 16 x 15 cm

Precisão:

-Eixos X e Y: 0.011mm

-Eixo Z: 0.025mm

Resolução de camada: 0.1mm

Diâmetro do filamento: 1.75mm

Compatibilidade do filamento: ABS, PLA, Flexível e Solúvel

Plataforma de impressão: alumínio aquecido

Dimensões do produto: 49 x 32 x 51.1 cm

Elétrica: 100-240V, ~4A, 50-60Hz, 24V DC @ 9.2A

Tipo de arquivo do software: .stl

Conectividade: USB, SD card

2.10.2.1 Configurações de impressão

Segue abaixo o passo a passo das configurações utilizadas. Lembrando que densidade maior, mais pesada a peça.

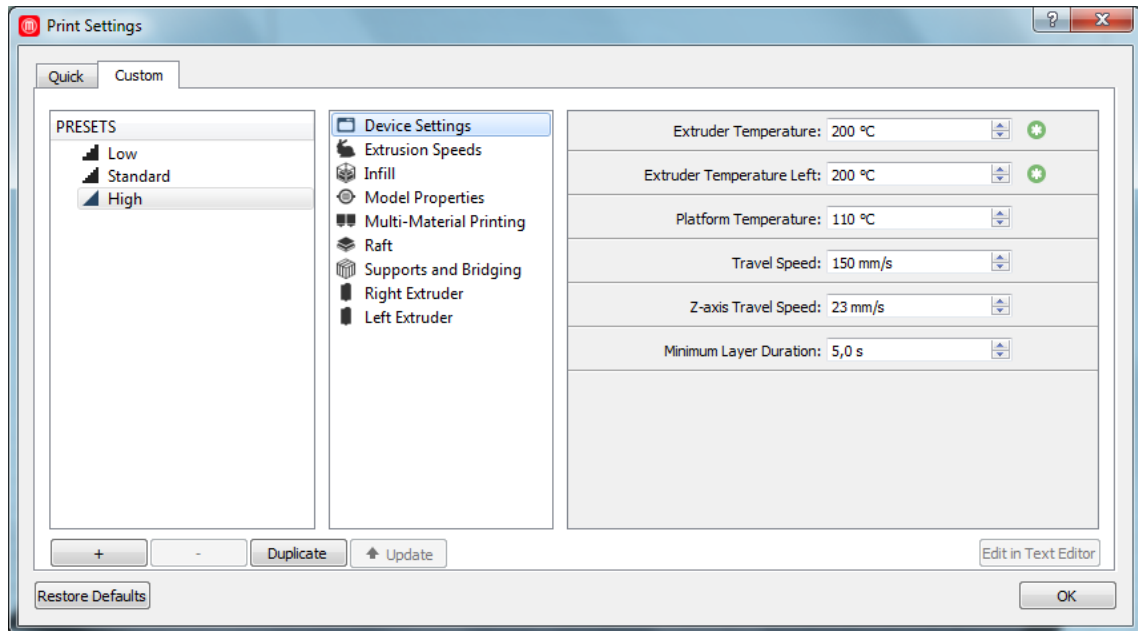


Figura 40: Configuração do dispositivo

Fonte: Autor

Não mudar nada em Extrusion Speeds.

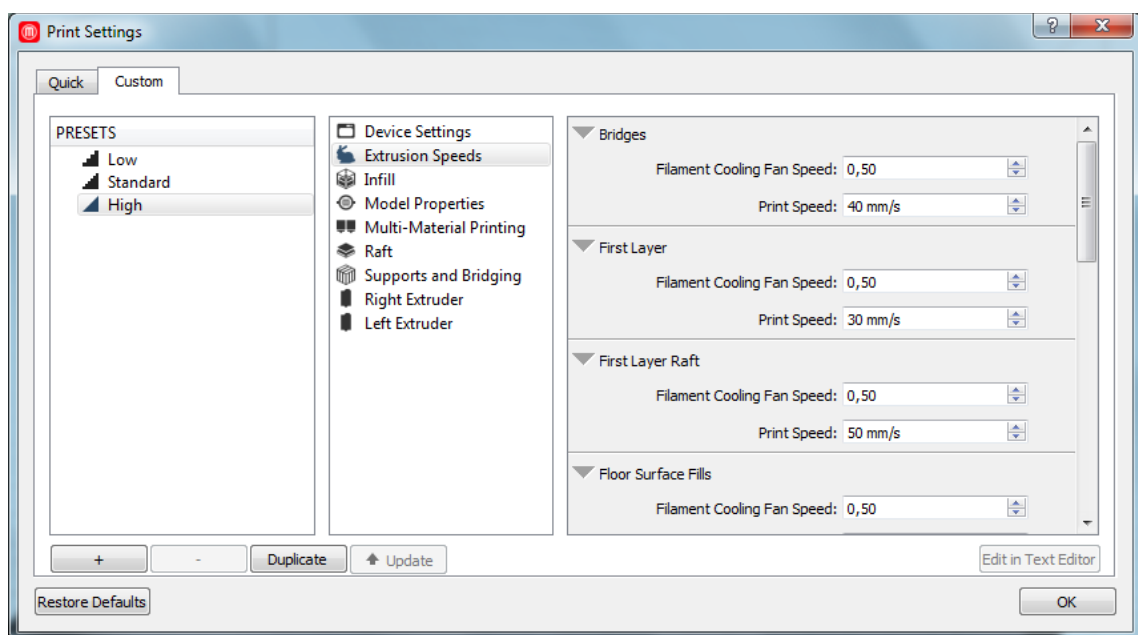


Figura 41: Extrusion Speeds

Fonte: Autor

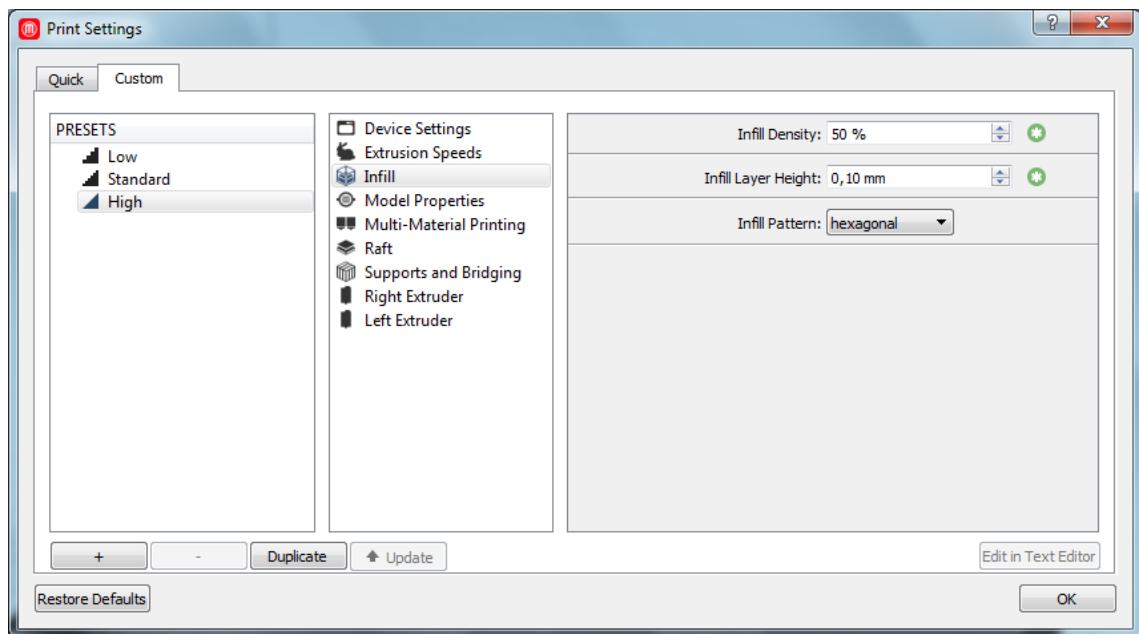


Figura 42: Infill (densidade)

Fonte: Autor

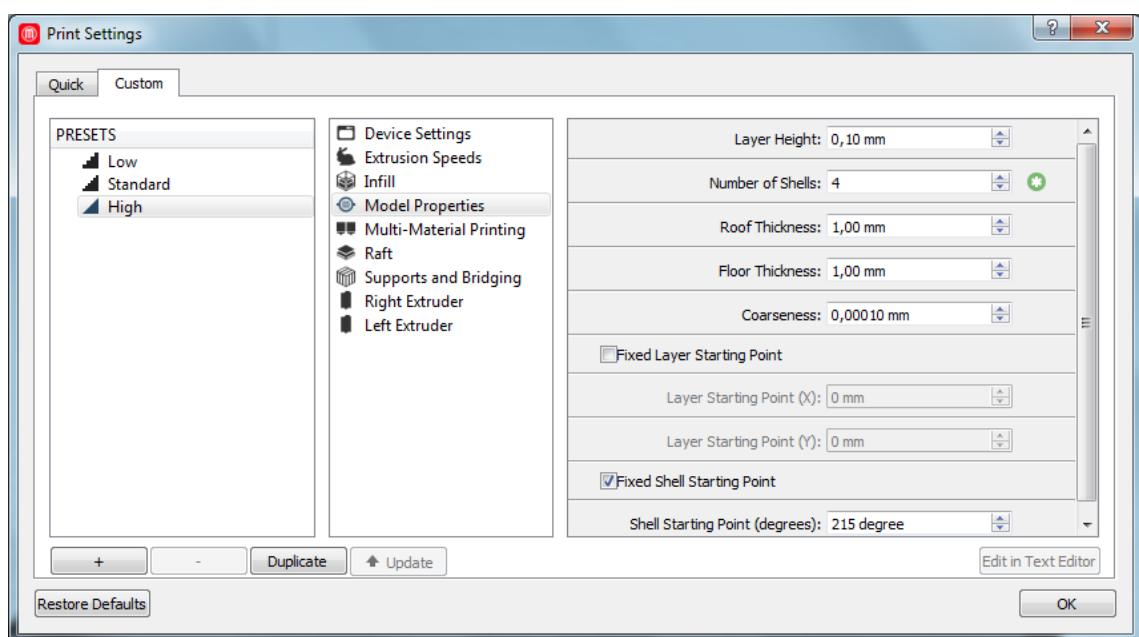


Figura 43: Propriedades do modelo

Fonte: Autor

Não mudar nada em Multi-Material Printing.

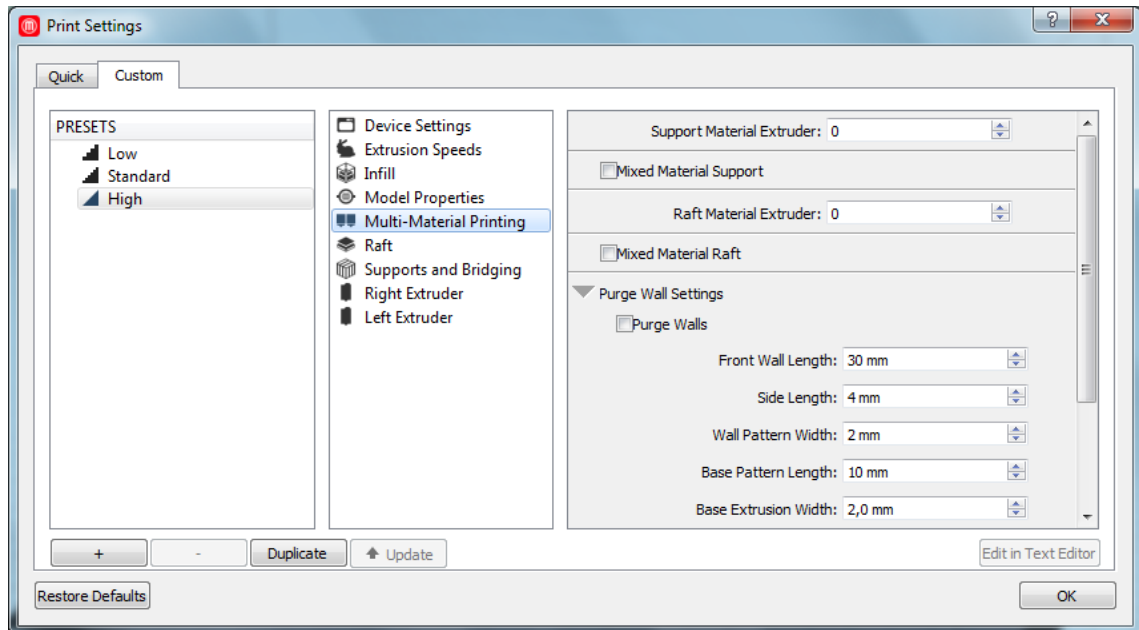


Figura 44: Multi-Material Printing

Fonte: Autor

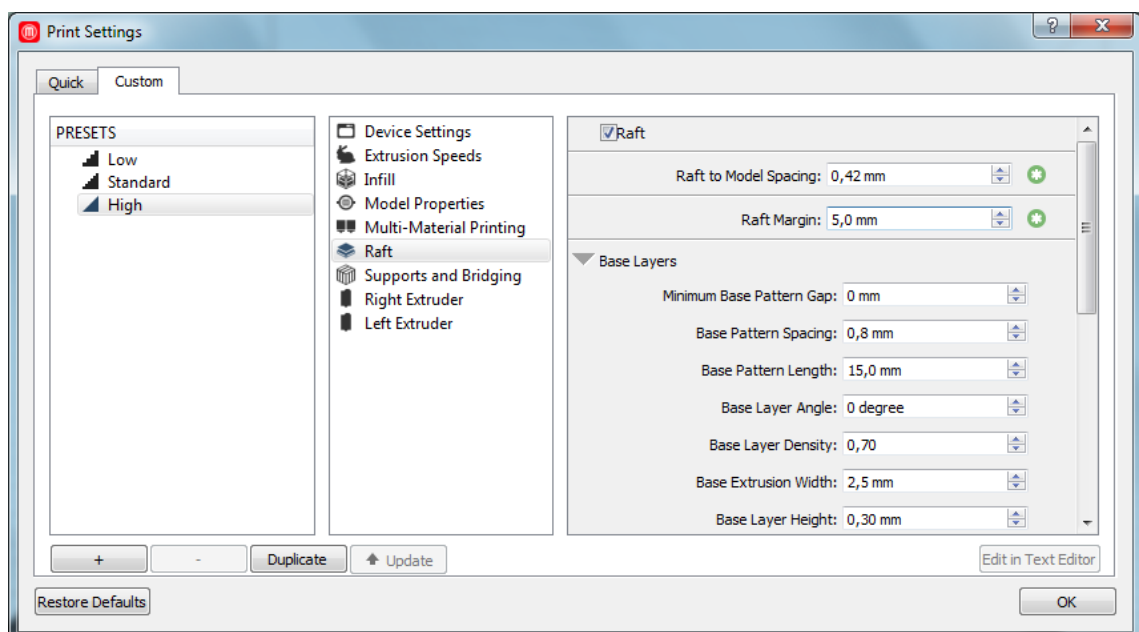


Figura 45: Raft

Fonte: Autor

Em Suportes, deve-se selecionar todos os ícones para não ter problemas com a impressão da peça.

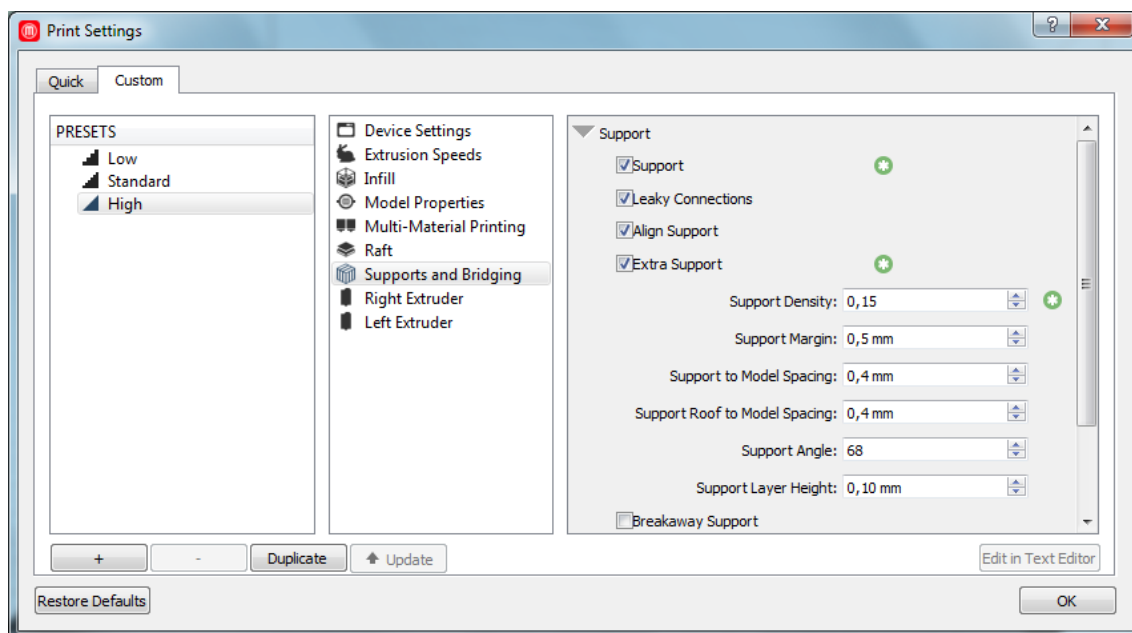


Figura 46: Supports and Bridging

Fonte: Autor

2. CONCLUSÃO

Como planejado o objetivo inicial foi alcançado, pois foi possível criar um controlador capaz de estabilizar a posição de uma câmera tendo como base as informações geradas por um IMU. O controle é estável e apesar de ter sido construído sobre um apoio de mão, poderia facilmente ser adaptado para outras bases ou dispositivos, desde quadricópteros para filmagens topográficas, até capacetes de ciclistas.

Porém, caso seja necessário a troca da câmera o controle precisaria ser alterado. Os ganhos do PID, a potência fornecida para os motores e talvez até a estrutura deverão se adaptar ao novo peso, centro de gravidade e até as novas dimensões da nova máquina.

Ainda assim, o projeto se mostrou capaz de controlar até uma garrafa plástica de água apenas apoiada na estrutura variando muito pouco os parâmetros do controlador. Portanto o dispositivo pode ser considerado também genérico.

Uma das dificuldades encontradas foram os motores escolhidos, os quais durante a execução do projeto descobrimos não serem adequados para este tipo de dispositivo, devido ao número de polos dos mesmos. E por isto, tivemos que comprar outros motores que no final foram eficazes para fazer o dispositivo funcionar.

É sugerido o aperfeiçoamento deste dispositivo, desenvolvendo uma estrutura que permita câmeras de dimensões maiores e com uma estrutura mais leve. Também seria interessante testar a resposta do controle em voos de drones, para garantir a total viabilidade do projeto. Poderiam também ser implementados outros métodos de estabilização, como o followmode do eixo Z, que faz com que a câmera acompanhe os movimentos horizontais, porém de forma suavizada, para evitar vibrações nas filmagens.

REFERÊNCIAS BIBLIOGRÁFICAS

- [1] WARFARE Technology. **“Gun Stabilization as Explained in 1944”**. Disponível em: <http://warfaretech.blogspot.com.br/2014/07/gun-stabilization-as-explained-in-1944.html>>. Acesso em: Jun/2016.
- [2] B. C. Nakra. **“Theory and Applications of Automatic Controls”**. Disponível em: <http://pdf-directory.org/ebook.php?id=ULGmKUazOOYC>>. Acesso em: Jun/2016.
- [3] C. D. Richard. **“Modern Control Systems”**. Disponível em: <http://dl.offdownload.ir/ali/Modern%20Control%20Systems.pdf>>. Acesso em: Jun/2016.
- [4] **“Motor Modeling and Position Control Lab”**. Disponível em: http://www.eng.buffalo.edu/~ms329/ProjectPages/MAE334_DynControlsMotorLab.pdf>. Acesso em: Jun/2016.
- [5] O. Oludayo. **“PID Control of Brushless DC Motor and Robot Trajectory Planning and Simulation with Matlab Simulink”**. Disponível em: <https://publications.theseus.fi/bitstream/handle/10024/7467/Oludayo%20Oguntoyinbo.pdf>>. Acesso em: Jun/2016.
- [6] EMBARCADOS. **“Sensores Inerciais”**. Disponível em: <http://www.embarcados.com.br/sensores-inerciais-parte-2/>>. Acesso em: Jun/2016.
- [7] EHOW. **“A diferença entre motores com e sem escovas”**. Disponível em: http://www.ehow.com.br/diferenca-entre-motores-escovas-info_48456/>. Acesso em: Jun/2016.
- [8] IFLIGHT. **“iPower GBM2208H-80T Gimbal Motor”**. Disponível em: <http://www.iflight-rc.com/ipower-gbm2208h-80t-gimbal-motor.html>>. Acesso em: Jun/2016.
- [9] BASECAM. **“BaseCam (AlexMos) SimpleBGC”**. Disponível em: <http://www.basecamelectronics.com/simplebgc/>>. Acesso em: Jun/2016.
- [10] DRONE Trest. **“How to connect and setup Alexmos 3 axis brushless gimbal controller”**. Disponível em: <http://www.dronetrest.com/t/how-to-connect-and-setup-alexmos-3-axis-brushless-gimbal-controller/53>>. Acesso em: Jun/2016.

[11] XOJO3D. **“Tutorial 12: Quaternions”**. Disponível em: <<http://www.xojo3d.com/tut012.php>>. Acesso em: Jun/2016.

[12] PLANNING Algorithms. **“Yaw, pitch, and roll rotations”**. Disponível em: <<http://planning.cs.uiuc.edu/node102.html>>. Acesso em: Jun/2016.

[13] STUDYWOLF. **“Robot Control Part 2: Jacobians, Velocity, and Force”**. Disponível em: <<https://studywolf.wordpress.com/2013/09/02/robot-control-jacobians-velocity-and-force/>>. Acesso em: Jun/2016.

[14] STARLINO. **“A Guide To using IMU (Accelerometer and Gyroscope Devices) in Embedded Applications”**. Disponível em: <http://www.starlino.com/imu_guide.html>. Acesso em: Jun/2016.

[15] WISHBOX Technologies. **“Impressoras 3D Makerbot”**. Disponível em: <<http://wishbox.net.br/produtos/impressao-3d/impressoras-3d-makerbot/>>. Acesso em: Jun/2016.

APÊNDICE A – CÓDIGO EM C

```
/* Includes -----*/

#include "stm32f1xx_hal.h"

#include "timer.h"

/* USER CODE BEGIN Includes */


/* USER CODE END Includes */


/* Private variables -----*/

SPI_HandleTypeDef hspi1;


TIM_HandleTypeDef htim1;
TIM_HandleTypeDef htim2;
TIM_HandleTypeDef htim3;


/* USER CODE BEGIN PV */

/* Private variables -----*/


/* USER CODE END PV */


/* Private function prototypes -----*/

void SystemClock_Config(void);
static void MX_GPIO_Init(void);
static void MX_SPI1_Init(void);
static void MX_TIM1_Init(void);
static void MX_TIM2_Init(void);
static void MX_TIM3_Init(void);


/* USER CODE BEGIN PFP */
```

```

/* Private function prototypes -----*/
void LED_Toggle(uint8_t LED);
/* USER CODE END PFP */

/* USER CODE BEGIN 0 */

/* USER CODE END 0 */

int main(void)
{

/* USER CODE BEGIN 1 */

/* USER CODE END 1 */

/* MCU Configuration-----*/

/* Reset of all peripherals, Initializes the Flash interface and the Systick. */
HAL_Init();

/* Configure the system clock */
SystemClock_Config();

/* Initialize all configured peripherals */
MX_GPIO_Init();
MX_SPI1_Init();
MX_TIM1_Init();
MX_TIM2_Init();
MX_TIM3_Init();

```

```

/* USER CODE BEGIN 2 */

TimerConfig();

/* USER CODE END 2 */


/* Infinite loop */
/* USER CODE BEGIN WHILE */
while (1)
{
/* USER CODE END WHILE */


    HAL_Delay(500);
    LED_Toggle(1);
    HAL_Delay(500);
    LED_Toggle(2);
/* USER CODE BEGIN 3 */


}

/* USER CODE END 3 */


}


/** System Clock Configuration
*/

void SystemClock_Config(void)
{

    RCC_OscInitTypeDef RCC_OscInitStruct;

```

```
RCC_ClkInitTypeDef RCC_ClkInitStruct;
```

```
RCC_OscInitStruct.OscillatorType = RCC_OSCILLATORTYPE_HSE;
```

```
RCC_OscInitStruct.HSEState = RCC_HSE_ON;
```

```
RCC_OscInitStruct.HSEPredivValue = RCC_HSE_PREDIV_DIV1;
```

```
RCC_OscInitStruct.PLL.PLLState = RCC_PLL_ON;
```

```
RCC_OscInitStruct.PLL.PLLSource = RCC_PLLSOURCE_HSE;
```

```
RCC_OscInitStruct.PLL.PLLMUL = RCC_PLL_MUL3;
```

```
HAL_RCC_OscConfig(&RCC_OscInitStruct);
```

```
RCC_ClkInitStruct.ClockType = RCC_CLOCKTYPE_SYSCLK;
```

```
RCC_ClkInitStruct.SYSCLKSource = RCC_SYSCLKSOURCE_PLLCLK;
```

```
RCC_ClkInitStruct.AHBCLKDivider = RCC_SYSCLK_DIV1;
```

```
RCC_ClkInitStruct.APB1CLKDivider = RCC_HCLK_DIV1;
```

```
RCC_ClkInitStruct.APB2CLKDivider = RCC_HCLK_DIV1;
```

```
HAL_RCC_ClockConfig(&RCC_ClkInitStruct, FLASH_LATENCY_0);
```

```
HAL_SYSTICK_Config(HAL_RCC_GetHCLKFreq()/1000);
```

```
HAL_SYSTICK_CLKSourceConfig(SYSTICK_CLKSOURCE_HCLK);
```

```
/* SysTick_IRQn interrupt configuration */
```

```
HAL_NVIC_SetPriority(SysTick_IRQn, 0, 0);
```

```
}
```

```
/* SPI1 init function */
```

```
void MX_SPI1_Init(void)
```

```
{
```

```

hspi1.Instance = SPI1;

hspi1.Init.Mode = SPI_MODE_MASTER;

hspi1.Init.Direction = SPI_DIRECTION_2LINES;

hspi1.Init.DataSize = SPI_DATASIZE_8BIT;

hspi1.Init.CLKPolarity = SPI_POLARITY_LOW;

hspi1.Init.CLKPhase = SPI_PHASE_1EDGE;

hspi1.Init.NSS = SPI_NSS_SOFT;

hspi1.Init.BaudRatePrescaler = SPI_BAUDRATEPRESCALER_2;

hspi1.Init.FirstBit = SPI_FIRSTBIT_MSB;

hspi1.Init.TIMode = SPI_TIMODE_DISABLED;

hspi1.Init.CRCCalculation = SPI_CRCCALCULATION_DISABLED;

hspi1.Init.CRCPolynomial = 10;

HAL_SPI_Init(&hspi1);

}

/* TIM1 init function */
void MX_TIM1_Init(void)
{

    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_BreakDeadTimeConfigTypeDef sBreakDeadTimeConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim1.Instance = TIM1;

    htim1.Init.Prescaler = 0;

    htim1.Init.CounterMode = TIM_COUNTERMODE_UP;

    htim1.Init.Period = 0;

    htim1.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;

```

```

htim1.Init.RepetitionCounter = 0;
HAL_TIM_PWM_Init(&htim1);

sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
HAL_TIMEx_MasterConfigSynchronization(&htim1, &sMasterConfig);

sBreakDeadTimeConfig.OffStateRunMode = TIM_OSSR_DISABLE;
sBreakDeadTimeConfig.OffStateIDLEMode = TIM_OSSI_DISABLE;
sBreakDeadTimeConfig.LockLevel = TIM_LOCKLEVEL_OFF;
sBreakDeadTimeConfig.DeadTime = 0;
sBreakDeadTimeConfig.BreakState = TIM_BREAK_DISABLE;
sBreakDeadTimeConfig.BreakPolarity = TIM_BREAKPOLARITY_HIGH;
sBreakDeadTimeConfigAutomaticOutput =
TIM_AUTOMATICOUTPUT_DISABLE;
HAL_TIMEx_ConfigBreakDeadTime(&htim1, &sBreakDeadTimeConfig);

sConfigOC.OCMode = TIM_OCMode_PWM1;
sConfigOC.Pulse = 0;
sConfigOC.OCpolarity = TIM_OCPolarity_HIGH;
sConfigOC.OCNPolarity = TIM_OCNPolarity_HIGH;
sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
sConfigOC.OCIdleState = TIM_OCIDLESTATE_RESET;
sConfigOC.OCNIdleState = TIM_OCNIDLESTATE_RESET;
HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_1);

HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_2);

HAL_TIM_PWM_ConfigChannel(&htim1, &sConfigOC, TIM_CHANNEL_3);

```



```

}

/* TIM2 init function */
void MX_TIM2_Init(void)
{

    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim2.Instance = TIM2;
    htim2.Init.Prescaler = 0;
    htim2.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim2.Init.Period = 0;
    htim2.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    HAL_TIM_PWM_Init(&htim2);

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    HAL_TIMEx_MasterConfigSynchronization(&htim2, &sMasterConfig);

    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_1);

    HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_2);

```

```

    HAL_TIM_PWM_ConfigChannel(&htim2, &sConfigOC, TIM_CHANNEL_3);

}

/* TIM3 init function */
void MX_TIM3_Init(void)
{

    TIM_MasterConfigTypeDef sMasterConfig;
    TIM_OC_InitTypeDef sConfigOC;

    htim3.Instance = TIM3;
    htim3.Init.Prescaler = 0;
    htim3.Init.CounterMode = TIM_COUNTERMODE_UP;
    htim3.Init.Period = 0;
    htim3.Init.ClockDivision = TIM_CLOCKDIVISION_DIV1;
    HAL_TIM_PWM_Init(&htim3);

    sMasterConfig.MasterOutputTrigger = TIM_TRGO_RESET;
    sMasterConfig.MasterSlaveMode = TIM_MASTERSLAVEMODE_DISABLE;
    HAL_TIMEx_MasterConfigSynchronization(&htim3, &sMasterConfig);

    sConfigOC.OCMode = TIM_OCMODE_PWM1;
    sConfigOC.Pulse = 0;
    sConfigOC.OCpolarity = TIM_OCPOLARITY_HIGH;
    sConfigOC.OCFastMode = TIM_OCFAST_DISABLE;
    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_1);

    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_2);

```

```

    HAL_TIM_PWM_ConfigChannel(&htim3, &sConfigOC, TIM_CHANNEL_3);

}

/** Configure pins as
    * Analog
    * Input
    * Output
    * EVENT_OUT
    * EXTI
*/
void MX_GPIO_Init(void)
{

    GPIO_InitTypeDef GPIO_InitStruct;

    /* GPIO Ports Clock Enable */
    __GPIOD_CLK_ENABLE();
    __GPIOA_CLK_ENABLE();
    __GPIOB_CLK_ENABLE();
    __GPIOC_CLK_ENABLE();

    /*Configure GPIO pins : LED1_Pin LED2_Pin */
    GPIO_InitStruct.Pin = LED1_Pin|LED2_Pin;
    GPIO_InitStruct.Mode = GPIO_MODE_OUTPUT_PP;
    GPIO_InitStruct.Pull = GPIO_PULLUP;
    GPIO_InitStruct.Speed = GPIO_SPEED_FREQ_HIGH;
    HAL_GPIO_Init(GPIOC, &GPIO_InitStruct);

```

```

}

#ifdef USE_FULL_ASSERT

void assert_failed(uint8_t* file, uint32_t line)
{
    /* USER CODE BEGIN 6 */
    /* User can add his own implementation to report the file name and line number,
       ex: printf("Wrong parameters value: file %s on line %d\r\n", file, line) */
    /* USER CODE END 6 */
}

#endif

void LED_Toggle(uint8_t LED){

    if(LED == 1){
        HAL_GPIO_TogglePin(GPIOC, LED1_Pin);
    }

    if(LED == 2){
        HAL_GPIO_TogglePin(GPIOC, LED2_Pin);
    }
}

```

APÊNDICE B – DESIGN ESTRUTURA

