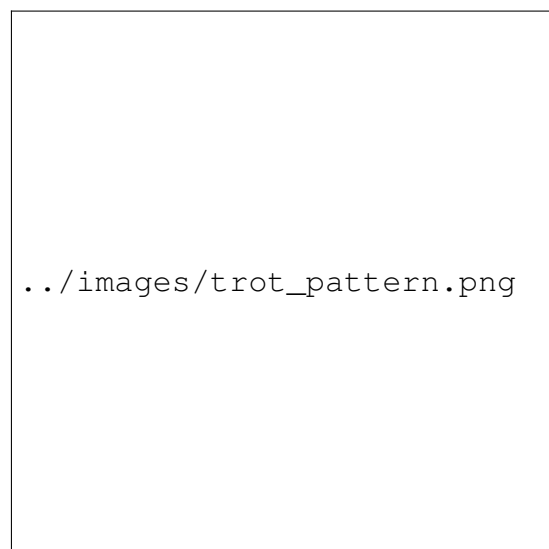


Baseado nos conceitos apresentados, foi projetado, simulado e desenvolvido um robô quadrúpede nomeado de Caramelo (figuras ??, ?? e ??). Caramelo é um robô quadrúpede de pequeno porte voltado para pesquisa e educação. Seu *hardware* foi modelado inteiramente pela equipe e impresso com impressora 3D no material ABS. Os atuadores do robô são servomotores do modelo *dynamixel* MX-28 e sua central de processamento é composta por uma RaspberryPi 4. Além disso, ele conta com um sensor inercial modelo MPU6050, que está instalado no corpo do robô. Este sensor contém um giroscópio e um acelerômetro, o que permite obter a aceleração linear, a velocidade angular e a orientação do corpo do robô. Todo o *software* foi desenvolvido com o ROS2 Humble (*Robot Operation System 2*) [??], que é um *framework* de robótica *open source* com vários recursos disponíveis para facilitar o desenvolvimento de sistemas robóticos.

A estrutura do caramelo é do tipo mamífero e a configuração das pernas é a *full-elbow*. Além disso, possui 3 GDL por perna, o que permite uma grande liberdade de movimentação para as patas. Seu *design* foi pensado para favorecer o balanço de massas entre o corpo e as pernas do robô, ou seja, a maior parte da massa se encontra no corpo ou próxima a ele. Os componentes eletrônicos internos, que abrangem sensores, unidades de processamento e a interface de comunicação com os atuadores, foram dispostos de forma simétrica, a fim de manter o centro de massa o mais próximo do centro do corpo. Os motores (componentes que contribuem com a maior massa para o sistema) foram dispostos o mais próximo possível do corpo. Um destaque para o motor que atua na junta da tíbia, é que foi instalado na parte de cima do fêmur com o objetivo de diminuir o momento de inércia da perna. Essa escolha demandou a adição de um sistema de transmissão entre o eixo do motor e a tíbia, formado por uma haste rígida de metal com duas juntas esfera nas extremidades.

A locomoção do Caramelo foi desenvolvida baseada nas marchas periódicas e simétricas. Dessa forma, foi adotada a marcha *trot* como a marcha principal do robô. Embora sua estrutura permita a realização de muitos outros tipos de marcha, neste trabalho, foi considerada apenas o *trot*, devido a sua simplicidade e eficiência. Com o objetivo de diminuir a complexidade do controle de locomoção, foi adotada uma marcha descontínua, ou seja, o corpo do robô se desloca apenas quando todas as patas estão do solo. A sequência de etapas da marcha do Caramelo pode ser vista na Figura ??, na qual as áreas em branco representam a etapa de *swing* e as em cinza a de *stance*. É possível perceber que sempre o mesmo par de pernas diagonais se move no mesmo instante. Entre duas etapas consecutivas de *swing*, há um momento em que todas as patas estão em *stance*, que é quanto o corpo do robô é deslocado no sentido desejado de locomoção.

Figura 1: Padrão de movimentação da marcha para cada perna.



Fonte: autores.

O sistema de controle do robô é composto por dois subsistemas principais: os controladores individuais de cada junta e os controladores da angulação do corpo do robô. Os controladores das juntas são os próprios controladores PID embarcados nos motores *dynamixel*. Foi utilizada a interface de controle

de posição com o atuador, de forma que o *setpoint* de controle enviado para cada motor é o ângulo em radianos para o qual ele deve rotacionar. O modelo cinemático do robô, apresentado na seção ??, é o responsável por mapear não apenas a posição tridimensional de cada pata com a angulação de cada junta, mas também a posição do corpo em seis dimensões: translação e rotação em  $(x, y, z)$ . Dessa forma, é possível controlar cada pata e o corpo do robô ao mesmo tempo de forma independente. Os controladores de angulação do corpo são dois controladores PID em paralelo, responsáveis por controlar o ângulo de *pitch* (rotação em  $y$ ) e o de *roll* (rotação em  $x$ ).

O planejador de marchas é o responsável por controlar cada pata do robô e, por consequência, o corpo. Ele calcula a trajetória que cada pata deve realizar, com base nas etapas de *stance* e *swing*, e envia o próximo ponto em que cada pata deve estar a uma frequência de 50Hz. Além disso, ele também considera o esforço de controle enviado pelos controladores de angulação, a fim de manter o corpo do robô em  $0^\circ$  a todo momento.

A seguir, serão apresentados o desenvolvimento do modelo cinemático, dos controladores de angulação e da trajetória que cada pata realiza na etapa de *swing*.

### 0.1 Modelo cinemático do Caramelo

Como dito anteriormente, o modelo cinemático é utilizado para resolver a cinemática inversa e a cinemática direta do robô. Para a cinemática direta, foi utilizado o pacote *tf2*, um recurso disponível no ROS2 que facilita o gerenciamento de transformações entre eixos de coordenadas. A cinemática inversa, por outro lado, foi feita com base em uma análise geométrica. As variáveis  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  expressam a posição angular de cada uma das juntas de uma perna do robô e são calculadas com auxílio das equações ?? a ?? em função da posição  $(x_{IK}, y_{IK}, z_{IK})$  desejada para a pata e dos comprimentos  $L_1$ ,  $L_2$  e  $L_3$  (figura ??).

$$\theta_1 = \arctan\left(\frac{x_{IK}}{y_{IK}}\right) - \arctan\left(\frac{L_1}{a}\right) \quad (1)$$

$$\theta_2 = \frac{\pi}{2} - \arctan\left(\frac{a}{z_{IK}}\right) - \arctan\left(\frac{\sqrt{1-A^2}}{A}\right) \quad (2)$$

$$\theta_3 = \arctan\left(\frac{\sqrt{1-B^2}}{B}\right) \quad (3)$$

$$a = \sqrt{x_{IK}^2 + y_{IK}^2 - L_1^2} \quad (4)$$

$$A = \frac{a^2 + z^2 + L_2^2 - L_3^2}{2L_2\sqrt{a^2 + z_{IK}^2}} \quad (5)$$

$$B = \frac{a^2 + z_{IK}^2 - L_2^2 - L_3^2}{2L_2L_3} \quad (6)$$

Figura 2: Links da perna do robô.



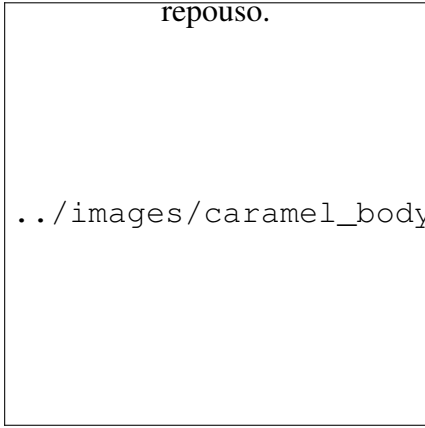
Fonte: autores.

Essas equações são úteis para o cálculo da posição de uma única perna, mas são insuficientes para realizar a cinemática do corpo do robô. Desta forma, um *frame* central, chamado de *base\_link* (figura ??), é utilizado como referência, e uma matriz  $T_M$  (eqs. ?? e ??) é utilizada para realizar a cinemática do corpo, a partir das translações  $(x_c, y_c, z_c)$  e rotações  $(\alpha, \beta, \gamma)$  desejadas, sendo possível controlar cada um dos 6 graus de liberdade. Para isso, as transformações  $T_{FR}$ ,  $T_{FL}$ ,  $T_{BL}$  e  $T_{BR}$  de cada um dos ombros (*hip links*) em relação ao *base\_link* são necessárias.

$$T_M = \begin{bmatrix} & & x_c \\ R_{xyz} & & y_c \\ & & z_c \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad (7)$$

$$R_{xyz} = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos \alpha & -\sin \alpha \\ 0 & \sin \alpha & \cos \alpha \end{bmatrix} \cdot \begin{bmatrix} \cos \beta & 0 & \sin \beta \\ 0 & 1 & 0 \\ -\sin \beta & 0 & \cos \beta \end{bmatrix} \cdot \begin{bmatrix} \cos \gamma & -\sin \gamma & 0 \\ \sin \gamma & \cos \gamma & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (8)$$

Figura 3: Eixos do robô em posição de repouso.



Fonte: autores.

O cálculo das angulações de cada perna então é feito utilizando como entrada os valores  $(x_{IK}, y_{IK}, z_{IK})$  resultantes de cada uma das transformações, conforme a equação ???. O mesmo cálculo é feito para as demais pernas, utilizando  $T_{FL}$ ,  $T_{BL}$  e  $T_{BR}$ .

$$\begin{bmatrix} x_{IK} \\ y_{IK} \\ z_{IK} \\ 1 \end{bmatrix} = (T_M \cdot T_{FR})^{-1} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (9)$$

Desta forma, a Cinemática Inversa é capaz de computar as angulações  $\theta_1$ ,  $\theta_2$  e  $\theta_3$  de cada uma das pernas a partir da posição  $(x, y, z)$  das patas em relação ao link central

do robô e às translações  $(x_c, y_c, z_c)$  e rotações  $(\alpha, \beta, \gamma)$  desejadas para o corpo. Entretanto, em muitos casos é mais conveniente realizar o cálculo dos ângulos passando como entrada as posições  $(x, y, z)$  das patas em relação à sua posição *default*, ou seja, a posição do seu link quando o robô está em seu estado de repouso (figura ??). Para isso, é possível realizar, previamente ao cálculo das angulações, mais uma transformação, desta vez do *base\_link* para cada uma das posições *default* das patas (equação ??).

$$\begin{bmatrix} x_{ik} \\ y_{ik} \\ z_{ik} \\ 1 \end{bmatrix} = (T_M \cdot T_{FR})^{-1} \cdot (F_{FR} \cdot \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix}) \quad (10)$$

## 0.2 Controle de angulação

Os controladores de angulação são dois controladores PID em paralelo responsáveis por controlar a rotação de *roll* e *pitch* do corpo do robô. Eles atuam de forma independente, controlando a rotação do corpo em ambos os eixos simultaneamente. Ambos os controladores são iguais e foram implementados seguindo o modelo apresentado no diagrama de blocos da figura ??.

O IMU é o sensor responsável por medir a rotação do corpo do robô, possibilitando a realimentação das saídas do sistema. O limitador foi adicionado para evitar que sejam enviados valores que extrapolam os limites de rotação das juntas do robô. Os esforços de controle são enviados o planejador de marchas que, por sua vez, envia os comandos de movimentação para os controladores das juntas.

## 0.3 Planejador de trajetória

O planejador de trajetória é responsável por calcular a trajetória que cada pata deve realizar, tanto na fase de *stance* quanto na fase de *swing*. Para o caramelo, a trajetória é uma curva cicloidal em ambas as etapas. Como apresentado em [??], uma curva

Figura 4: Controlador PID projetado.



Fonte: autores

cicloidal pode ser definida no espaço tridimensional entre os pontos  $(x_o, y_o, z_o)$  e  $(x_f, y_f, z_f)$  em função do tempo  $t$  pelas equações (??) a (??), sendo  $H$  a altura do passo e  $T$  o período.

$$x = (x_f - x_o) \frac{K - \sin(K)}{2\pi} + x_o \quad (11)$$

$$y = (y_f - y_o) \frac{K - \sin(K)}{2\pi} + y_o \quad (12)$$

$$z = H \frac{1 - \cos(K)}{2} + z_o \quad (13)$$

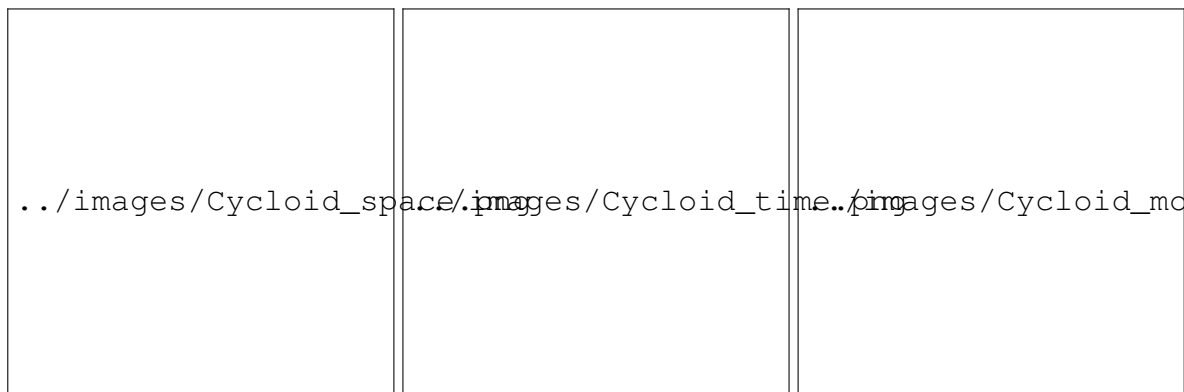
$$K = \frac{2\pi t}{T} \quad (14)$$

O gráfico de uma curva cicloidal no espaço 3D pode ser vista na Figura ???. A mesma trajetória é ilustrada na Figura ?? em função do tempo. É possível perceber que a curva possui primeira derivada nula no momento em que a pata toca o solo, o que é favorável ao controle de malha aberta já que quanto mais suave for a aterrissagem, menos distúrbios são causados no sistema.

Além da altura, distâncias em  $x$  e  $y$  e período, o planejador de trajetória do Caramelo também conta com dois parâmetros que têm como objetivo melhorar ainda mais o controle da força com que a pata toca o chão.

Como os atuadores do robô são servomotores controlados por posição, o torque é proporcional ao deslocamento que este deve realizar entre os pontos da trajetória. Ou seja, quanto maior a resolução da trajetória, mais suave será o movimento. No entanto, a resolução da trajetória  $N$  é fixa, dada em função do período do passo e da frequência de controle do planejador de marchas (50 Hz)  $N = 50T$ . Logo, a estratégia adotada é a de espaçar a mesma quantidade de pontos de forma desigual ao longo do período do passo, de forma que haja mais pontos próximos ao momento em que a pata aterrissa no solo e menos pontos próximos ao momento em que ele é erguido. O parâmetro  $P_T$  é uma fração do período total do passo e o parâmetro  $P_N$  a fração do número total de pontos do passo que deve se encontrar entre o tempo 0 e  $P_T \cdot T$ . Em outras palavras, se  $P_T = 0.66$  e  $P_N = 0.33$ , 33% de  $N$  estará nos primeiro dois terços do período, enquanto os 67% restantes estarão no um terço final. A trajetória considerando esses parâmetros está ilustrada na Figura ??.

Figura 5: Trajetórias cicloidais para o passo de robô.



(a) Curva no espaço 3D.

(b) Curva no tempo.

(c) Curva com disposição de pontos modificada.

Fonte: autores.