

Sistema de Controle por Joystick - Plataforma de Stewart

📋 Visão Geral

Sistema completo de controle em tempo real da Plataforma de Stewart usando gamepad/joystick, integrado ao backend FastAPI e frontend Three.js.

🎮 Características

- **Controle em Tempo Real:** Mapeamento direto dos eixos do joystick para translação/rotação da plataforma
- **Limites de Segurança:** $\pm 30\text{mm}$ de translação e $\pm 8^\circ$ de rotação (configuráveis)
- **Preview 3D em Tempo Real:** Visualização instantânea da pose calculada
- **Modo Dual:** Preview-only ou aplicação no hardware
- **Validação Automática:** Verifica limites dos atuadores antes de aplicar

🕹️ Mapeamento do Controle

Stick Esquerdo

- **Horizontal (LX):** Translação em X ($\pm 30\text{mm}$)
- **Vertical (LY):** Translação em Y ($\pm 30\text{mm}$)

Stick Direito

- **Horizontal (RX):** Rotação Pitch ($\pm 8^\circ$)
- **Vertical (RY):** Rotação Roll ($\pm 8^\circ$)

Triggers (Futuro)

- **LT/RT:** não implementado ainda

Zona Morta

- Valores menores que 0.1 (10%) são ignorados para evitar drift

🏗️ Arquitetura

Backend ([app.py](#))

Novo Modelo Pydantic

```
class JoystickPoseRequest(BaseModel):  
    lx: float = Field(0.0, ge=-1.0, le=1.0) # Stick esquerdo X  
    ly: float = Field(0.0, ge=-1.0, le=1.0) # Stick esquerdo Y  
    rx: float = Field(0.0, ge=-1.0, le=1.0) # Stick direito X
```

```

    ry: float = Field(0.0, ge=-1.0, le=1.0) # Stick direito Y
    lt: Optional[float] = None # Trigger esquerdo
    rt: Optional[float] = None # Trigger direito
    apply: bool = False # Se True, aplica no hardware
    z_base: Optional[float] = None # Z base (default = h0)

```

Endpoint: POST /joystick/pose

Funcionalidades:

1. Recebe eixos normalizados do joystick (-1..1)
2. Mapeia para valores físicos (mm e graus)
3. Calcula cinemática inversa
4. Valida limites dos atuadores
5. Se `apply=True` e válido, envia comando serial `spmm6x=...`
6. Retorna resposta completa com pose, comprimentos, pontos 3D

Limites Físicos:

```

MAX_TRANS_MM = 30.0 # ±30mm
MAX_ANGLE_DEG = 30.0 # ±30°

```

Exemplo de Request:

```
{
  "lx": 0.5,
  "ly": -0.3,
  "rx": 0.2,
  "ry": -0.1,
  "apply": true,
  "z_base": 432
}
```

Exemplo de Response (válida):

```
{
  "valid": true,
  "applied": true,
  "pose": {
    "x": 5.0,
    "y": 3.0,
    "z": 432,
    "roll": 1.0,
    "pitch": 2.0,
    "yaw": 0.0
  },
  "comprimento": 10.0
}
```

```

    "lengths_abs": [ 590.2, 588.5, ...],
    "course_mm": [ 90.2, 88.5, ...],
    "base_points": [[ 305.5, -17, 0], ...],
    "platform_points": [[ 196.1, -238.5, 432], ...]
}

```

Frontend

joystick-control.js

Classe Principal: JoystickController

```

class JoystickController {
  constructor(config)
  setEnabled(enabled)          // Ativa/desativa controle
  setApplyToHardware(apply)     // Define se aplica no hardware
  getState()                   // Retorna estado atual
  destroy()                     // Limpa recursos
}

```

Configuração:

```

const JOYSTICK_CONFIG = {
  DEADZONE: 0.1, // Zona morta (10%)
  UPDATE_RATE_MS: 50, // Taxa de envio ao backend (20Hz)
  PREVIEW_RATE_MS: 16, // Taxa de atualização 3D (~60fps)
  MAX_TRANS_MM: 10.0,
  MAX_ANGLE_DEG: 10.0,
  Z_BASE: null, // null = usar h0 do backend
};

```

Loops Independentes:

1. **Loop de Preview** (`requestAnimationFrame`): Lê gamepad e atualiza UI/3D a ~60fps
2. **Loop de Backend** (`setInterval`): Envia comandos ao servidor a cada 50ms

controller.js

Script principal da página que:

- Inicializa preview 3D (Three.js)
- Cria instância do `JoystickController`
- Gerencia UI (checkboxes, valores, sliders)
- Conecta callbacks (`onPoseChange`, `onError`)
- Gerencia eventos de gamepad (conectar/desconectar)

controller.html

Página completa com:

- Header institucional (IFSP)
- Navegação entre páginas
- Status de conexão serial
- Painel de controle do joystick
- Instruções de uso
- Display de valores atuais (X, Y, Z, Roll, Pitch, Yaw)
- Preview 3D em tempo real

💡 Como Usar

1. Iniciar Backend

```
cd interface/backend  
python app.py
```

Backend rodará em <http://localhost:8001>

2. Abrir Frontend

Abra <interface/frontend/controller.html> no navegador

3. Conectar Porta Serial

1. Clique em "Atualizar" para listar portas
2. Selecione a porta do ESP32
3. Clique em "Conectar"

4. Conectar Gamepad

1. Conecte um gamepad USB ou Bluetooth
2. O sistema detectará automaticamente
3. Status mudará para "Conectado: [nome do gamepad]"

5. Ativar Controle

1. Marque "Ativar Controle por Joystick"
2. Mova os sticks - o preview 3D atualizará em tempo real
3. Valores de X, Y, Z, Roll, Pitch, Yaw serão exibidos

6. Aplicar no Hardware (Opcional)

⚠️ ATENÇÃO: Só faça isso se a plataforma estiver segura!

1. Marque "Aplicar no Hardware"
2. Movimentos do joystick agora controlam a plataforma real
3. Comandos `spmm6x=...` serão enviados via serial

🛡️ Segurança

Limites Físicos Implementados

- **Translação X, Y:** $\pm 30\text{mm}$ (configurável)
- **Rotação Roll, Pitch, Yaw:** $\pm 8^\circ$ (configurável)
- **Z:** Fixo em 500mm por padrão

Validação em Múltiplas Camadas

1. **Frontend:** Clamp de valores antes de enviar
2. **Backend:** Clamp novamente + validação de cinemática inversa
3. **ESP32:** Limites de curso (500-680mm)

Zona Morta

- Valores $< 10\%$ são zerados
- Previne drift e movimentos não intencionais

🔧 Configurações Avançadas

Ajustar Limites

Edite em `app.py`:

```
MAX_TRANS_MM = 30.0
MAX_ANGLE_DEG = 8.0
```

Ajustar Taxa de Atualização

Edite em `joystick-control.js`:

```
UPDATE_RATE_MS: 50, // 20Hz (mais rápido = 30ms, mais lento = 100ms)
```

Ajustar Zona Morta

```
DEADZONE: 0.1, // 10% (aumentar para 0.15 se houver drift)
```

💡 Troubleshooting

Gamepad não detectado

1. Certifique-se de que o gamepad está conectado
2. Pressione qualquer botão para "acordar" o gamepad
3. Verifique no console do navegador se há mensagens de conexão

4. Teste em outro navegador (Chrome recomendado)

Preview 3D não atualiza

1. Verifique se o backend está rodando
2. Abra o console e procure por erros
3. Verifique se a URL da API está correta (<http://localhost:8001>)

Comandos não aplicados no hardware

1. Certifique-se de que a serial está conectada
2. Verifique se "Aplicar no Hardware" está marcado
3. Verifique no console do backend se há mensagens de TX serial
4. A pose pode ser inválida (fora dos limites) - veja logs

Movimentos muito sensíveis/insensíveis

Ajuste a zona morta ou os limites máximos conforme descrito acima



Autor

Guilherme Miyata - Instituto Federal de São Paulo (IFSP)

Trabalho de Conclusão de Curso - 2025

[Github](#)

[Linkedin](#)

[Portfólio](#)

Última atualização: Novembro 2025