Integração de Robótica Móvel e Manipulação Industrial

Sincronização entre Pioneer P3-DX e TM5-900 via ROS

Equipe de Desenvolvimento

Universidade Federal de Itajubá
Engenharia de Controle e Automação

10 de dezembro de 2025

Roteiro

O Desafio: Integração de Ilhas de Produção

Cenário

Em ambientes industriais, robôs móveis (AMRs) e braços manipuladores operam isolados. A transferência de material exige sincronização precisa para evitar colisões e garantir eficiência.

Objetivos do Projeto:

- **Centralização:** Criar um nó "Maestro"para orquestrar a operação.
- **Navegação:** Levar o Pioneer da prateleira até a mesa C4 autonomamente.
- **Manipulação:** Comandar o braço TM5 para classificar a peça.

Arquitetura da Solução

O sistema adota uma topologia centralizada para contornar limitações de hardware dos robôs.

1. Subsistema Móvel (Pioneer)

- Comunicação via Wi-Fi.
- Hardware antigo: Atua apenas como "escravo"(Drivers).

2. Subsistema Manipulador (TM5)

- Conexão Ethernet/Wi-Fi.
- Controle Híbrido: Nó ROS + Script Embarcado (Listen Node).

Distribuição de Processamento (Quem roda o quê?)

A carga computacional foi dividida para garantir performance em tempo real:

Notebook (Workstation) - O "Cérebro"

Aqui roda o processamento pesado e a lógica:

- `roscore`: O Master do sistema.
- `maestro.py`: Nó principal que contem a Máquina de Estados e o Plano de Produção.
- `move_base`: Calcula rotas (Global/Local Planner) e desvia de obstáculos.
- `rviz`: Visualização do mapa e laser.

Pioneer P3-DX (No Robô)

Apenas drivers de baixo nível:

- `RosAria`: Comunicação com motores/encoders.
- `laser`: Leitura do Lidar.

TechMan TM5 (No Braço)

- `Listen Node`: Script interno aguardando comandos do Tópico ROS.

Fluxograma de Execução (Maestro.py)

A lógica sequencial garante a segurança da operação:

1. **Início:** Leitura do Plano de Produção (YAML).
2. **Navegação:** Pioneer transporta a peça até a mesa **C4**.
3. **Handshake (Sincronia):**
   - Maestro publica comando no tópico `/comando_robo`.
   - Maestro entra em *Sleep* (espera calculada) enquanto o braço opera.
4. **Manipulação (Pick & Place):**
   - Braço: `PEGAR` $\rightarrow$ `DESTINO` $\rightarrow$ `HOME`.
5. **Liberação:** O Pioneer só recebe permissão para sair após o tempo de segurança do braço expirar.

Desafios Técnicos Enfrentados

1. Latência e Watchdog (O Robô "Travando")

Problema: O Pioneer parava abruptamente ("soluços") pois o delay do Wi-Fi.
Solução: Centralização do `roscore` no notebook e aumento do timeout do driver `RosAria` para 2.0s.

2. "Ressaca Temporal"(TF_OLD_DATA)

Problema: Rede isolada sem NTP causou divergência de relógios. O ROS rejeitava dados do Laser (Erro de Extrapolação).
Solução: Ajuste manual via `date` e aumento do `transform_tolerance` para 20.0s no Costmap.

3. MoveIt (Graus vs Radianos)

Problema: O planejador recebia ângulos em graus (ex: 90) mas interpretava como radianos (giros infinitos).
Solução: Implementação de conversão `math.radians()` no código.

Resultados e Conclusão

- **Funcionamento Pleno:** Ciclo completo (Busca $\rightarrow$ Entrega $\rightarrow$ Manipulação) realizado autonomamente.
- **Arquitetura Robusta:** A decisão de rodar o *Master* no Notebook foi crucial para estabilizar a navegação em rede instável.
- **Segurança:** A lógica sequencial impediu colisões entre os robôs.

Obrigado!

Dúvidas?