

Sistema distribuído baseado em ROS 2 para controle de robôs móveis aplicados ao futebol

Augusto Ribeiro Castro
Orientador: Prof. Dr. Valdir Grassi Junior

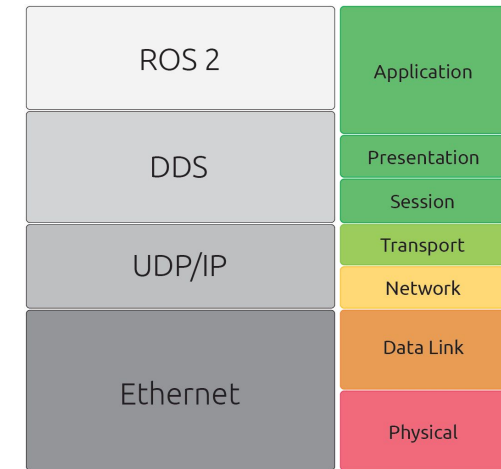
04 de dezembro de 2020

- Robôs móveis:
 - Se movem de uma forma autônoma de um ponto a outro;
 - Buscam um conjunto de objetivos;
 - Comumente utilizados em times:
 - Aumenta a cobertura;
 - Aumenta a tolerância a falhas.
- *Robot World Cup (RoboCup)*:
 - Desenvolvimento da robótica e da inteligência artificial:
 - Ambiente do futebol de robôs;
 - Warthog Robotics possui trabalhos prévios na área:
 - Alternativa: *Robot Operating System* (ROS);
 - ROS 2:
 - Sistemas de tempo real;
 - Redes não ideais;
 - Pequenas plataformas embarcadas.

- *Middleware* que conecta clusters de computação heterogênea;
- Grafo composto por vértices e arestas:
 - Vértices (**nós**): programas ou processos;
 - Arestas: comunicação entre os nós;
- **Nós:**
 - Publicam/recebem **mensagens** em **tópicos** de forma assíncrona:
- **Serviços:**
 - Comunicações síncronas entre os **nós** por meio de um par de mensagens;
 - São interações ocasionais e de tempo de execução específico.
- **Parâmetros:**
 - Variáveis do **nó** que podem ser modificadas durante o tempo de execução;

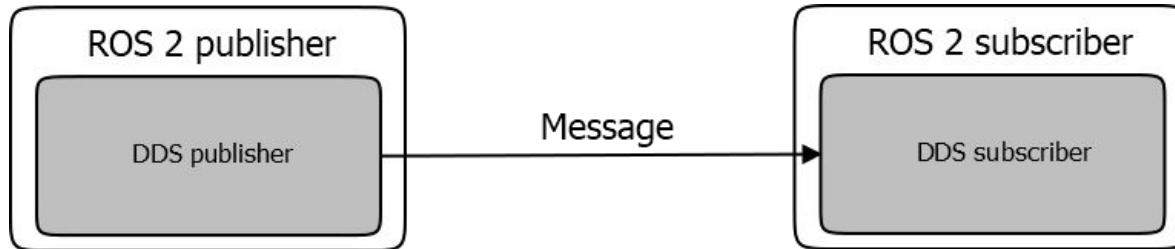
- Baseadas no padrão Data Distribution Service (DDS):
 - Modelo para sistemas centrados nos dados;
 - O DDS também é um middleware!
 - A comunicação do ROS 2 é construída a partir das funcionalidades do DDS:
 - Elimina o *roscore*!
 - Habilita o conceito de QoS!

Representação do ROS 2 no modelo OSI



Gutierrez et al. (2018)

Relação entre os pares *publisher/subscriber* do DDS e do ROS 2

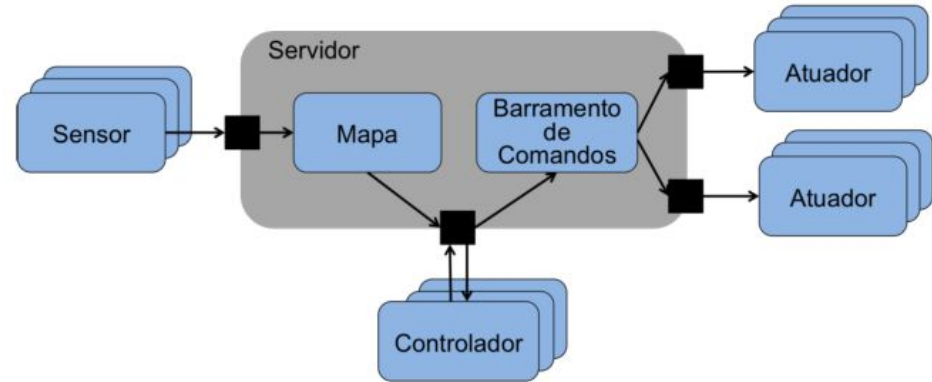


Elaborada pelo autor

- Políticas de comunicação entre os nós:
 - Criação de protocolos tão confiáveis quanto o TCP ou tão eficientes quanto o UDP, mas com diversos estados intermediários entre esses extremos;
 - Configuração de parâmetros como o tempo máximo de validade de uma mensagem, tempo máximo para uma mensagem ser publicada num tópico, confiabilidade do envio, tamanho da fila, tempo de inatividade de um nó e a volatilidade de um tópico;
 - O ROS 2 fornece perfis padrões de comportamento:
 - Dados de sensores;
 - Serviços;
 - Parâmetros;
 - ROS 1.

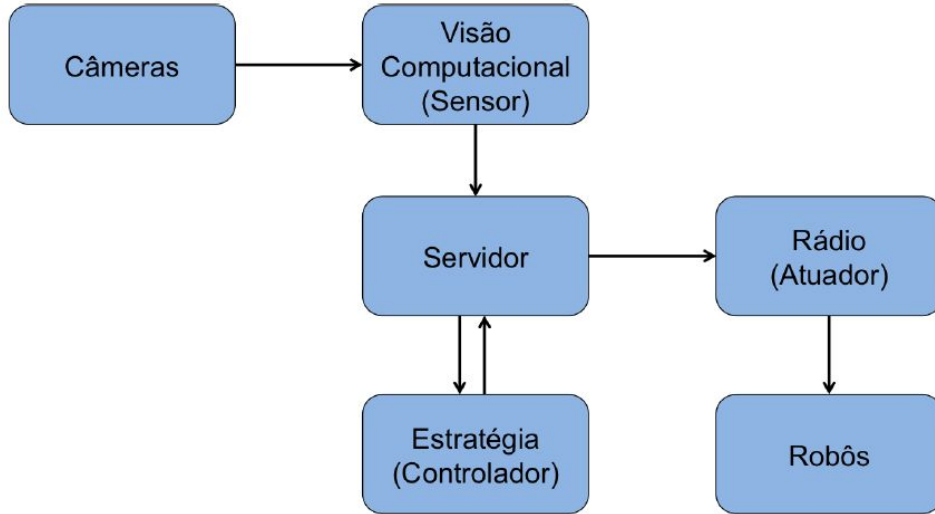
- Proposta por Lang (2015);
- Base para o controle de robôs heterogêneos;
- Composta por seis módulos bem definidos:
 - **Sensor;**
 - **Servidor:**
 - Mapa;
 - Barramento de comandos.
 - **Controlador;**
 - **Atuador.**

Módulos da arquitetura GEARSystem



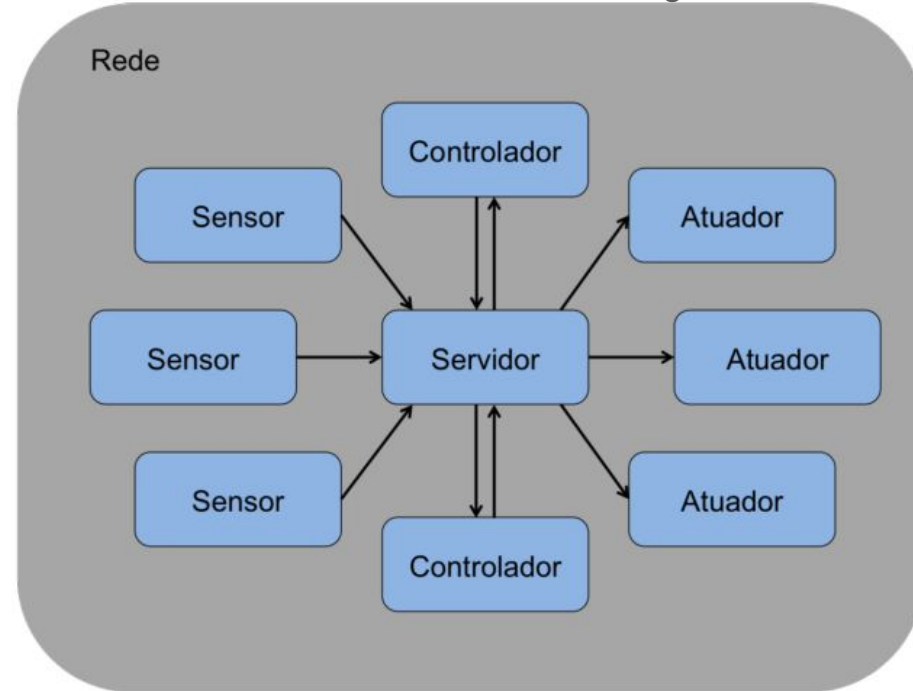
Lang (2015)

Exemplo no futebol de robôs



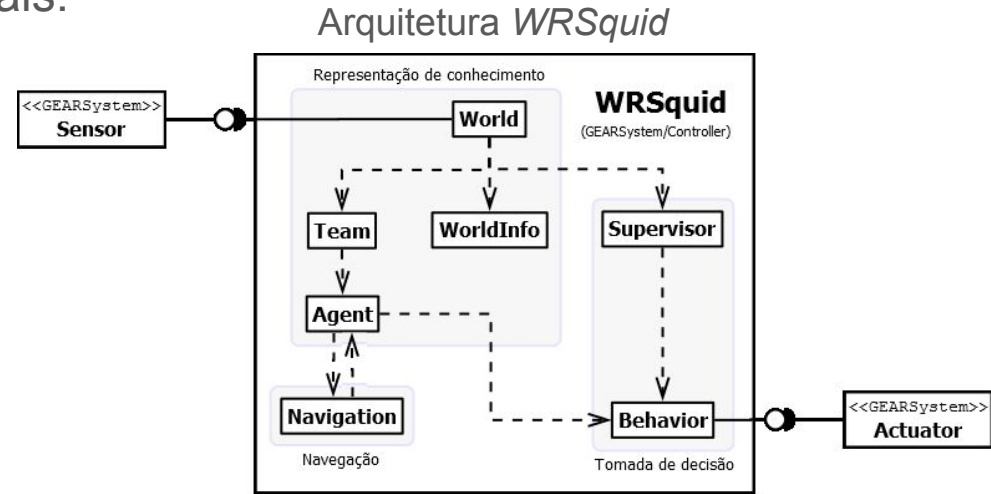
Lang (2015)

Sistema multi-robótico heterogêneo



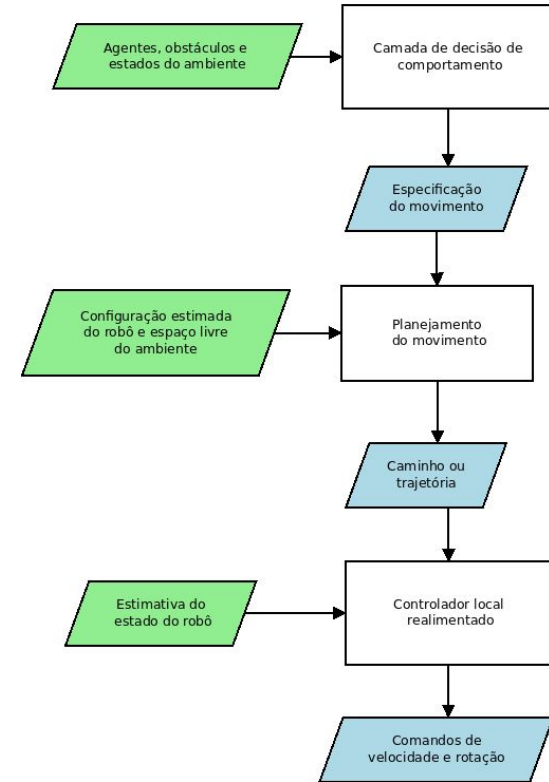
Lang (2015)

- Proposta por Oliveira (2017);
- Estende o controlador da arquitetura GEARSystem;
- Atribuição de comportamentos a robôs heterogêneos;
- Composta por três módulos principais:
 - Representação de conhecimento;
 - Tomada de decisão;
 - Navegação.



Oliveira (2017)

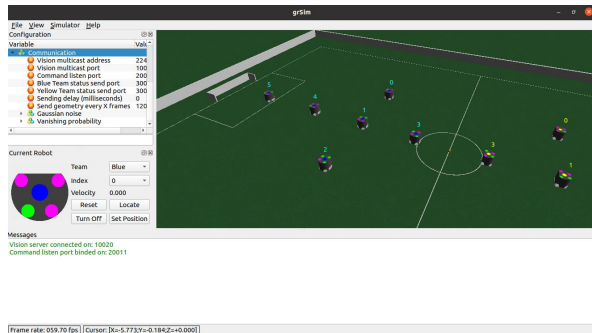
- Foi estudado o fluxograma ao lado, baseado no trabalho de Paden et al. (2016):
 - Camada de decisão de comportamentos:** sistemas fuzzy, redes neurais artificiais, algoritmos genéticos e aprendizado por reforço;
 - Planejamento do movimento:** algoritmos baseados em campos potenciais artificiais ou em buscas em grafos;
 - Controlador local realimentado:** compensadores PI/PID, sistemas fuzzy.



Paden et al. (2016)

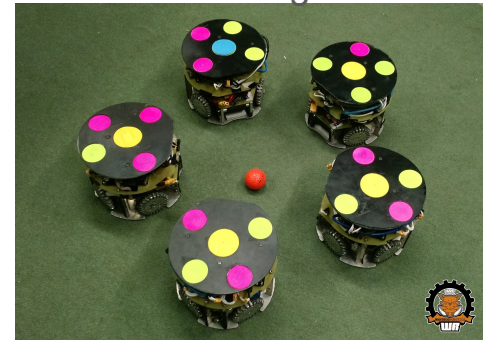
- ROS 2 Foxy Fitzroy lançado em junho de 2020 e com suporte até maio de 2023;
- Ubuntu 20.04, biblioteca rclcpp e framework Qt;
- Divisão B da categoria RoboCup Small Size League (SSL).

Interface do simulador grSim



Elaborada pelo autor

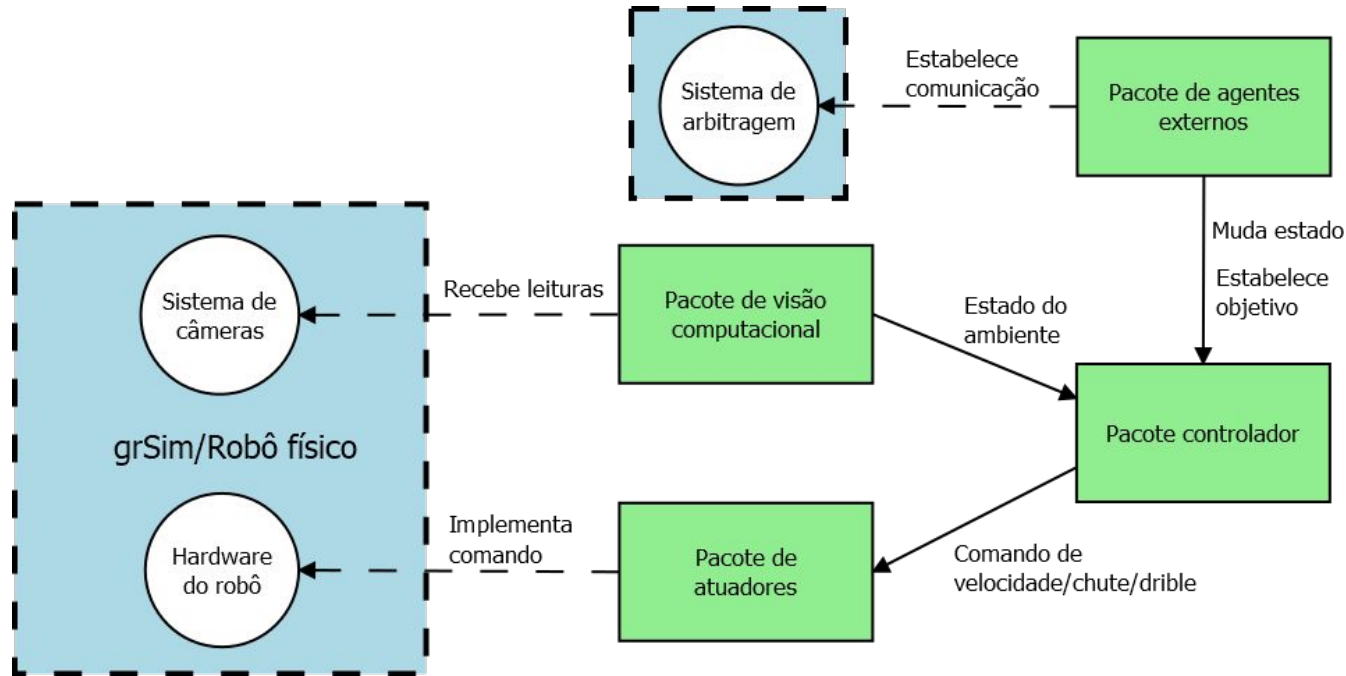
Robôs da categoria SSL



Oliveira (2017)

- Processamento distribuído:
 - Facilitaria o processo de embarcar todo o processamento dos robôs num futuro próximo;
 - Tomada de decisão feita localmente:
 - Informar outros agentes por meio de mensagens;
 - Câmeras fixas em cada robô ao invés de câmeras no teto.

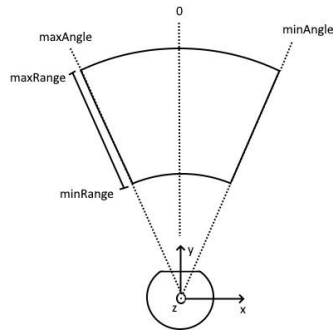
Domínios do sistema proposto



Elaborada pelo autor

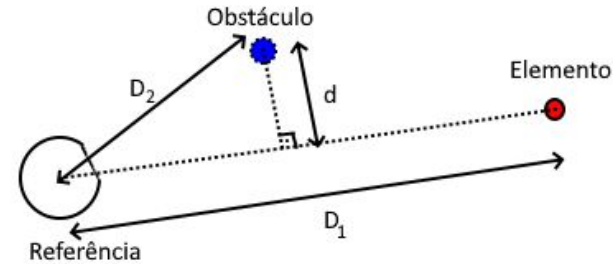
- Abriga nós que interpretam o ambiente por meio de sensores:
 - Publicam mensagens informando sobre os elementos detectados:
 - Posições, em metros, de bolas e robôs detectados;
 - Orientação, id e os times de robôs detectados;
 - Medida de confiança das posições.
- Implementação para o grSim:

Cálculo de campo visual



Elaborada pelo autor

Cálculo de oclusão



Elaborada pelo autor

- Trata entradas externas que modificam o funcionamento da estrutura de tomada de decisão dos robôs;
- Entrada do software de arbitragem:
 - Envio de uma mensagem de estado;
 - Outros exemplos: testes que determinam posição de jogadores, requisitam uma jogada específica, modificam um comportamento...

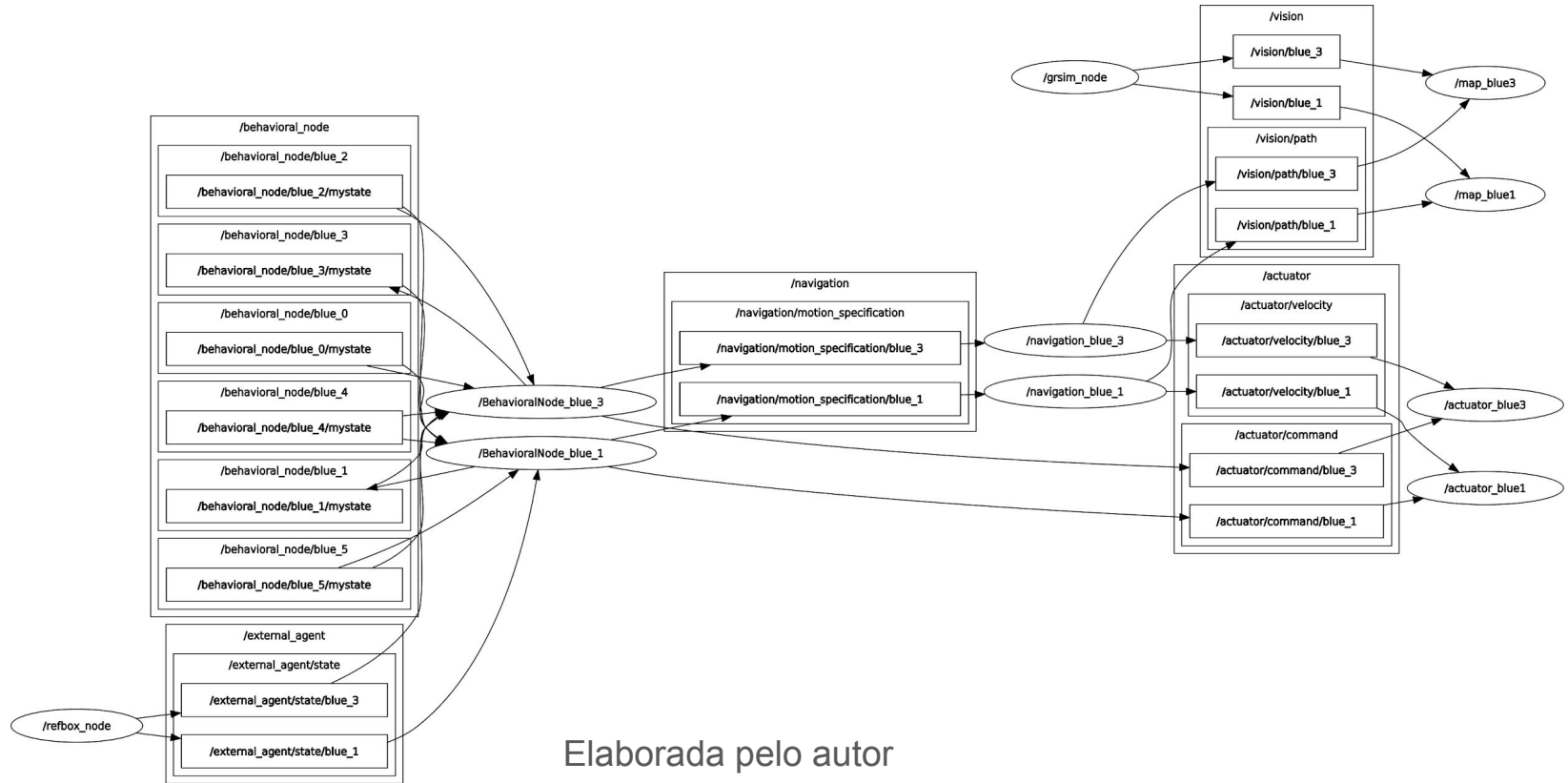
- Implementa a estrutura de tomada de decisão estudada:
 - Três nós:
 - Mapa;
 - Nó de comportamento;
 - Nó de navegação.

- Sobrescreve ao tópico de visão computacional:
 - Abriga os dados de forma coerente e segura.
- Fornece serviços a outros nós:
 - Informações de elementos e do campo;
 - **InfoBus:**
 - Classe auxiliar;
 - Consultas de alto nível de abstração.
- Implementação pelo usuário:
 - Estrutura estática de informações sobre o campo;
- Abriga a interface gráfica do sistema.

- Utiliza o **InfoBus** e o tópico do agente externo;
- Define um estado atual e aos outros robôs:
 - O estado escolhido tem como objetivo acionar os dispositivos do atuador e especificar movimentos ao nó de navegação:
 - Utiliza métodos como `goTo()`, `goToLookTo()`, `lookTo()`, `holdBall()` e `kick()`;
- Classe abstrata para o nó:
 - Interfaces e requisitos do ROS 2;
 - Utiliza o padrão de projeto orientado a objeto **State**:
 - Usuário implementa estados:
 - Execução do estado;
 - Define qual o próximo estado;
 - Invoca a transição;
 - A transição (alocação/deleção/informação do estado) é tratada pelo projeto.

- Trata as interfaces e requisitos do ROS 2;
- Fornece acesso à **InfoBus**;
- Usuário implementa dois tipos algoritmos:
 - **Algoritmo de navegação:**
 - *calculatePath()*;
 - *checkCurrentPath()*;
 - *updatePathTracking()*.
 - **Algoritmos de controles:**
 - Método que itera sobre uma variável de erro da malha de controle;
 - Função auxiliar para declaração de parâmetros.

- Fornece nós que recebem mensagens padrões do ROS para transformá-las em comandos para o robô:
 - Recebe comandos de velocidade (linear e angular);
 - Recebe comandos para acionamento de dispositivos (dispositivo de drible e chute);
- Classe abstrata para facilitar a implementação de nós de atuadores:
 - Tratamento de interfaces e requisitos do ROS 2;
 - Usuário implementa apenas dois métodos para efetivar o comando em *hardware*.



Elaborada pelo autor

Elementos visíveis para $minAngle = -45^\circ$, $maxAngle = 45^\circ$, $minRange = 0\text{ m}$, $maxRange = 20\text{ m}$ e $d = 180\text{ mm}$ para dois robôs ou $d = 90\text{ mm}$ para a bola:



Elaborada pelo autor

(a) Nó do mapa para o robô 1 do time azul.



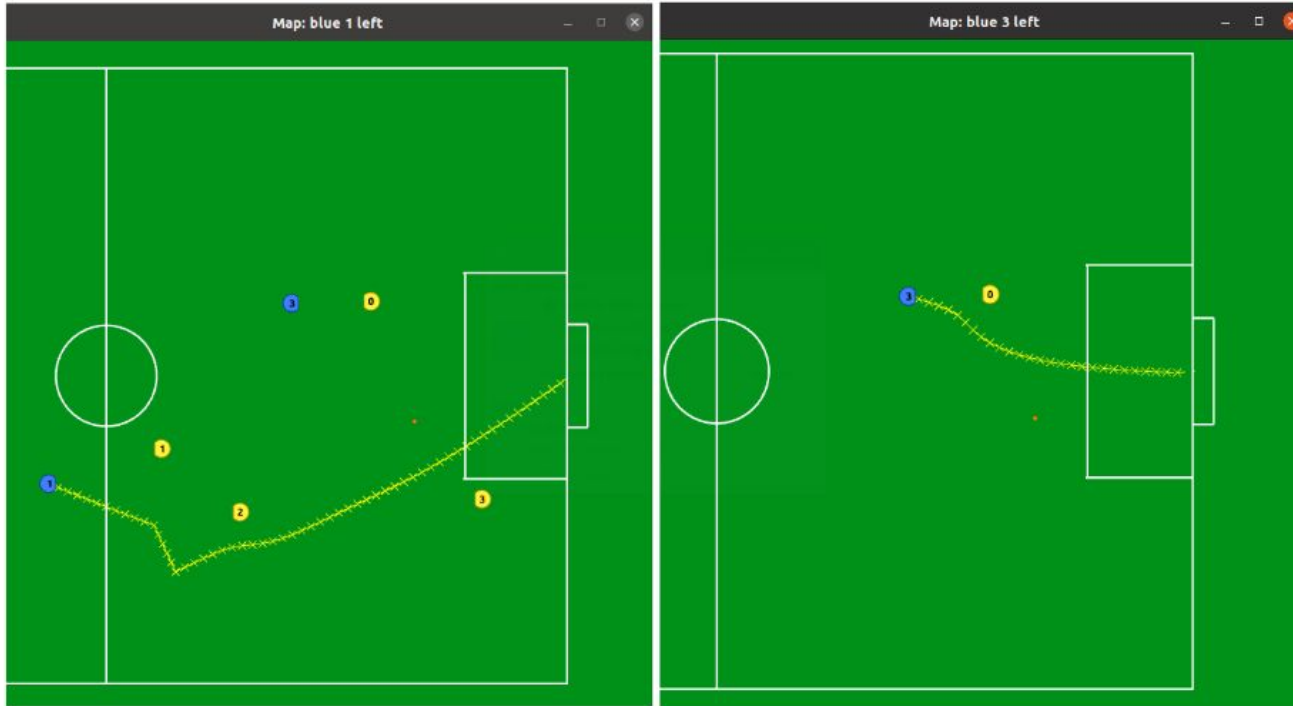
(b) Nó do mapa para o robô 3 do time azul.



Elaborada pelo autor

(a) Nó do mapa para o robô 1 do time azul.

(b) Nó do mapa para o robô 3 do time azul.



Elaborada pelo autor

- Abaixo são mostradas características dos perfis padrões fornecidos pelo ROS 2:
 - Serviços e parâmetros: mantêm os perfis padrões do ROS 2;
 - Demais interfaces: proposição de perfis específicos.

Comparação entre características de diferentes perfis QoS

Perfil de QoS	Histórico	Tamanho da fila	Confiabilidade
Padrão (ROS 1)	<i>keep last</i>	10	Garante o recebimento de todos os pacotes
<i>Sensor data</i>	<i>keep last</i>	5	Tenta enviar todos os pacotes, mas perdas podem acontecer devido à rede
Serviços	<i>keep last</i>	10	Garante o recebimento de todos os pacotes
Parâmetros	<i>keep last</i>	1000	Garante o recebimento de todos os pacotes

Elaborada pelo autor

- Interface **/vision/marcador:**
 - Exemplo claro de dados de sensores:

Atraso e idade das mensagens para 10000 amostras

Perfil de QoS	Atraso (s)				Idade das mensagens (ms)			
	Médio	Min.	Máx.	Desvio padrão	Média	Mín.	Máx.	Desvio padrão
Padrão (ROS 1)	0,001	0,000	0,004	0,00022	0,0895	0,0	118,0	2,3276
<i>Sensor data</i>	0,001	0,000	0,007	0,00029	0,0289	0,0	8,0	0,2547

Elaborada pelo autor

- Envio comandos à navegação e ao atuador:
 - Priorizar o recebimento de dados mais recentes;
 - Execução contínua.

- Envio de estados:
 - Ocorre apenas quando uma transição ocorre (não determinístico);
 - A confiabilidade é vital dado que os estados estão ligadas às regras do jogo.
- Perfil para enviar mensagens de estados:
 - Fila de mensagens mais longa que o perfil padrão;
 - Maior segurança em casos de oscilações da rede.

Características do perfil proposto para o envio de estados

Perfil de QoS	Histórico	Tamanho da fila	Confiabilidade
Perfil de estados	<i>keep last</i>	1000	Garante o recebimento de todos os pacotes

Elaborada pelo autor

- Resultados médios (atraso e idade das mensagens) semelhantes:
 - Máximos relativamente altos provavelmente devidos a oscilações na rede;
 - Impossível medir a perda de pacotes;
 - O aumento da fila é benéfico quando é preciso lidar com oscilações da rede e o armazenamento necessário pode ser fornecido.

Atraso e idade das mensagens para 10000 amostras

Perfil de QoS	Atraso (s)				Idade das mensagens (ms)			
	Médio	Min.	Máx.	Desvio padrão	Média	Mín.	Máx.	Desvio padrão
Padrão (ROS 1)	0,001	0,000	0,016	0,00062	0,1164	0,0	11,0	0,5155
Perfil de estados	0,001	0,000	0,010	0,00057	0,1644	0,0	132,0	1,8607

Elaborada pelo autor

- Apresenta uma base teórica a respeito do ROS 2:
 - Útil para *Warthog Robotics*;
 - Importante para o autor.
- O sistema pode ser aplicado à categoria RoboCup *Small Size League*:
 - Representação de conhecimento, geração de caminhos e desempenho de troca de mensagens;
 - Possibilidade de adaptá-lo para outras categorias.
- Corroborar com Scheunemann e Dijk (2019) quanto ao uso do ROS 2 na RoboCup;
- Melhorias futuras:
 - Mais estados e utilização de robôs reais;
 - Teste de componentes (memória compartilhada);
 - Explorar sistemas de tempo real e tolerância a falhas.

Referências

- BEOM, H. R.; KOH, K. C.; CHO, H. S. Behavioral control in mobile robot navigation using fuzzy decision making approach. In: IEEE.Proceedings of IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'94). [S.I.],1994. v. 3, p. 1938–1945.
- BRUCE, J. R.Real-time motion planning and safe navigation in dynamic multi-robot environments. [S.I.], 2006.
- EVERETT, M.; CHEN, Y. F.; HOW, J. P. Motion planning among dynamic,decision-making agents with deep reinforcement learning. In: IEEE.2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). [S.I.],2018. p. 3052–3059.
- FU, Z.; CHEN, Y.; DING, Y.; HE, D. Pollution source localization based on multi-uav cooperative communication.IEEE Access, v. 7, p. 29304–29312, 2019.
- GUTIERREZ, C. S. V.; JUAN, L. U. S.; UGARTE, I. Z.; VILCHES, V. M.Towards a distributed and real-time framework for robots: Evaluation of ROS 2.0communications for real-time robotic applications. 2018.
- JOSEPH, L. ROS Robotics Projects. [S.I.]: Packt Publishing Ltd, 2017.

- KHATIB, O. Real-time obstacle avoidance for manipulators and mobile robots. In: Autonomous robot vehicles. [S.l.]: Springer, 1986. p. 396–404.
- KIM, J.; SMEREKA, J. M.; CHEUNG, C.; NEPAL, S.; GROBLER, M. Security and performance considerations in ros 2: a balancing act. arXiv preprint arXiv:1809.09566, 2018.
- KITANO, H.; ASADA, M.; NODA, I.; MATSUBARA, H. Robocup: robot world cup. IEEE Robotics Automation Magazine, v. 5, n. 3, p. 30–36, 1998.
- LANG, R. G. Desenvolvimento de Arquitetura de Controle Distribuído de Sistemas Multi-Robóticos. 2015. 79 p. Dissertação (Mestrado em Engenharia Elétrica)— Escola de Engenharia de São Carlos, Universidade de São Paulo, 2015.
- LANG, R. G.; SILVA, I. N. d.; ROMERO, R. A. F. Development of distributed control architecture for multi-robot systems. In: 2014 Joint Conference on Robotics: SBR-LARS Robotics Symposium and Robocontrol. [S.l.: s.n.], 2014. p. 163–168.
- LEE, M. Evolution of behaviors in autonomous robot using artificial neural network and genetic algorithm. Information Sciences, Elsevier, v. 155, n. 1-2, p. 43–60, 2003.

- LEWIS, F. L.; GE, S. S. Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications. [S.l.]: CRC Press, 2006.
- LI, S.; GUO, Y.; BINGHAM, B. Multi-robot cooperative control for monitoring and tracking dynamic plumes. In: 2014 IEEE International Conference on Robotics and Automation (ICRA). [S.l.: s.n.], 2014. p. 67–73.
- LIM, Y.; CHOI, S.-H.; KIM, J.-H.; KIM, D.-H. Evolutionary univector field-based navigation with collision avoidance for mobile robot. IFAC Proceedings Volumes, Elsevier, v. 41, n. 2, p. 12787–12792, 2008.
- LIMA, P. U.; CUSTODIO, L. M. M. Artificial intelligence and systems theory: Applied to cooperative robots. International Journal of Advanced Robotic Systems, v. 1, n. 3, p. 15, 2004. Disponível em: <<https://doi.org/10.5772/5630>>.
- MONAJJEMI, V.; KOOCHAKZADEH, A.; GHIDARY, S. S. grsim – robocup small size robot soccer simulator. In: RÖFER, T.; MAYER, N. M.; SAVAGE, J.; SARANLI, U. (Ed.). RoboCup 2011: Robot Soccer World Cup XV. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. p. 450–460. ISBN 978-3-642-32060-6.

- OLIVEIRA, G. C. de. Arquitetura baseada em comportamentos para controle de múltiplos robôs heterogêneos com aplicação no futebol de robôs. In: . [S.l.: s.n.], 2017. p. 78.
- OMG, O. M. G.Data Distribution Service (DDS), Version 1.4. [S.I.], 2015.Disponível em: <<https://www.omg.org/spec/DDS/1.4>>.
- PADEN, B.; CAP, M.; YONG, S. Z.; YERSHOV, D.; FRAZZOLI, E.A Survey of Motion Planning and Control Techniques for Self-driving Urban Vehicles. 2016.
- PADHY, P. K.; SASAKI, T.; NAKAMURA, S.; HASHIMOTO, H. Modeling and position control of mobile robot. In: IEEE.2010 11th IEEE International Workshop on Advanced Motion Control (AMC). [S.I.], 2010. p. 100–105.
- QUIGLEY, M.; CONLEY, K.; GERKEY, B.; FAUST, J.; FOOTE, T.; LEIBS, J.;WHEELER, R.; NG, A. Y. Ros: an open-source robot operating system. In: KOBE,JAPAN.ICRA workshop on open source software. [S.I.], 2009. v. 3, n. 3.2, p. 5.
- QUIGLEY, M.; GERKEY, B.; SMART, W. D.Programming Robots with ROS: a practical introduction to the Robot Operating System. [S.I.]: "O'Reilly Media,Inc.", 2015.

- ROS 2.About Quality of Service settings. [S.I.], 2020. Disponível em: <<https://index.ros.org/doc/ros2/Concepts/About-Quality-of-Service-Settings/>>.
- ROS 2. Composing multiple nodes in a single process. [S.I.], 2020. Disponível em: <<https://index.ros.org/doc/ros2/Tutorials/Composition/>>.
- SCHEUNEMANN, M. M.; DIJK, S. G. van. Ros 2 for robocup. In: SPRINGER.Robot World Cup. [S.I.], 2019. p. 429–438.
- SCHNEIER, M.; BOSTELMAN, R.Literature review of mobile robots for manufacturing. [S.I.]: US Department of Commerce, National Institute of Standards and Technology, 2015.
- SHEN, Z.; HE, Z.; LI, S.; WANG, Q.; SHAO, Z. A multi-quadcopter cooperative cyber-physical system for timely air pollution localization. ACM Trans. Embed.Comput. Syst., Association for Computing Machinery, New York, NY, USA, v. 16, n. 3,abr. 2017. ISSN 1539-9087. Disponível em: <<https://doi.org/10.1145/3005716>>.

- SIEGWART, R.; NOURBAKHSH, I. R. Introduction to autonomous mobile robots.[S.l.]: MIT press, 2004. 453 p. ISBN 9780262195027.
- SILVA, M. O. d. Campos potenciais modificados aplicados ao controle de múltiplos robôs. 2011. Tese (Doutorado) — Universidade de São Paulo, 2011.