

# Arquitetura com Computação na Borda e Controle Preditivo Baseado em Modelo para Controle de Veículos Autoguiados

Rômulo A. L. V. de Omena\* Danilo F. S. Santos\* Angelo Perkusich\*

\* Núcleo de Pesquisa, Desenvolvimento e Inovação - VIRTUS,  
Universidade Federal de Campina Grande, Campina Grande-PB,  
Brasil (e-mails: {romulo.omena, danilo.santos,  
perkusic}@embedded.ufcg.edu.br).

**Abstract:** Automated Guided Vehicles (AGVs) are essential for industry material transportation. In the Industry 4.0 and Industrial Internet of Things scenario, edge computing combined with new generation networks may support the navigation control of AGVs. Centralizing on edge not only facilitates the integration of the control system with the other factory systems but also reduces the cost of vehicles and battery consumption once the edge computing allocates the tasks which require more computing power. From the results of previous experiments, we found that Model-Based Predictive Control, together with edge computing and wireless networks, is a robust control solution for AGVs. The Model Predictive Control's predictive nature can keep vehicles stable even in delays or packet loss on the network. Besides that, this approach promotes free navigation without fixed paths, which reduces costs and facilitates layout changes. Therefore, based on previous results and literature review, we propose an architecture for AGVs control with edge computing using the Model Predictive Control in this paper. The proposed architecture can support the AGVs navigation on smart factories of Industry 4.0, also in cases of network signal degradation.

**Resumo:** Os Veículos Autoguiados ou AGVs (*Automated Guided Vehicles*) são essenciais na indústria para o transporte de materiais. No contexto da Indústria 4.0 e da Internet Industrial das Coisas, a computação na borda em conjunto com redes sem fio de nova geração podem dar suporte no controle da navegação dos AGVs. A centralização na borda não só facilita a integração do sistema de controle com os demais sistemas da fábrica, mas também reduz o custo dos veículos e o consumo de bateria, dado que a computação na borda aloca as tarefas que requerem mais poder computacional. A partir dos resultados de experimentos anteriores, verificamos que o Controle Preditivo Baseado em Modelo, juntamente com computação na borda e redes sem fio, é uma solução de controle robusta para AGVs. A natureza preditiva do Controle Preditivo Baseado em Modelo pode manter os veículos estáveis mesmo em situações de atrasos ou perdas de pacotes na rede. Além disso, essa abordagem promove a navegação livre, sem rotas fixas, o que reduz custos e facilita a alteração de layouts. Portanto, baseado nos resultados anteriores e na revisão bibliográfica, propomos neste artigo uma arquitetura para controle de AGVs com computação na borda, utilizando o Controle Preditivo Baseado em Modelo. A arquitetura proposta pode dar suporte na navegação de AGVs nas fábricas inteligentes da Indústria 4.0, inclusive quando houver degradação do sinal na rede.

*Keywords:* AGV; Edge Computing; Model Predictive Control; Industry 4.0; IIoT.

*Palavras-chaves:* AGV; Computação na Borda; Controle Preditivo Baseado em Modelo; Indústria 4.0; IIoT.

## 1. INTRODUÇÃO

Na era da Internet das Coisas (IoT – *Internet of Things*) a computação está espalhada em inúmeros dispositivos conectados à Internet e está invisivelmente embutida nos ambientes ao nosso redor (Gubbi et al., 2013), com aplicações distribuídas em diversos domínios, tais como, agricultura, saúde, transporte, indústria, compras, educação, veículos e casas inteligentes (Al-Fuqaha et al., 2015). Na indústria, o conceito da IoT se estende para a Internet Industrial das Coisas (IIoT – *Industrial Internet of Things*), onde as

“coisas” industriais, como sensores, atuadores e máquinas estão conectadas à Internet (Aazam et al., 2018).

Nas aplicações de IIoT, os recursos de computação em nuvem podem falhar no atendimento aos requisitos temporais, de segurança e confiabilidade, quando dados em massa são enviados para processamento, que no caso das fábricas inteligentes podem alcançar o nível de Gigabytes por segundo (Qiu et al., 2020). Em operações sensíveis ao tempo, como por exemplo, a parada de emergência de uma máquina ou a frenagem de emergência de um veículo,

graves consequências podem ocorrer se os dados forem enviados para a nuvem e uma resposta não for recebida em tempo hábil (Qiu et al., 2020). Uma solução para o problema é a transferência dos dados para os recursos computacionais na borda da rede, o que caracteriza a computação na borda (Mao et al., 2017). Nesse modelo computacional, os dados são analisados e processados utilizando os recursos computacionais e de rede distribuídos nos caminhos entre as fontes de dados e as plataformas de computação em nuvem (Qiu et al., 2020).

Do ponto de vista de rede, a quinta geração de comunicações móveis, o 5G, na qual conexões ultra-confiáveis e de baixa latência podem ser disponibilizadas, é uma importante forma de acesso à computação na borda através das redes de acesso por rádio (RAN – *Radio Access Networks*) (Kiani and Ansari, 2018). Esse método de acesso é denominado *Multi-Access Edge Computing* (MEC) (Qiu et al., 2020). Redes com diferentes características poderão ser disponibilizadas no 5G. As redes uRLLC (*Ultra-Reliable and Low-Latency Communications*), por exemplo, são utilizadas em aplicações que exigem conexão ultra-confiável e de baixa latência. Os serviços uRLLC têm relevância para a Indústria 4.0, dado que níveis baixos e determinísticos de latência são geralmente demandados (García-Morales et al., 2019).

Uma aplicação que pode ser explorada nesse cenário de controladores baseados em software (Zhao and Dán, 2018) com computação na borda e redes 5G, são os Veículos Autoguiados (AGVs – *Automated Guided Vehicles*), tipos de robôs móveis utilizados para transporte de materiais em ambientes industriais. Se uma fábrica já tem um sistema de comunicação 5G instalado, o mesmo sistema pode então ser utilizado para o gerenciamento da frota de AGVs, controle, integração com o estoque e análises em tempo real (Oyekanlu et al., 2020). Ao usar um sistema de MEC com 5G, os dados dos AGVs podem ser coletados e alguns dos veículos podem ser alocados com base no volume de trabalho e no estado da produção, auxiliando na redução de custos e dinamizando a manufatura (Oyekanlu et al., 2020).

Nesse contexto de computação distribuída e redes sem fio de próxima geração, neste trabalho é proposta uma arquitetura com computação na borda e Controle Preditivo Baseado em Modelo para o controle de AGVs industriais. Essa abordagem de controle tem se mostrado uma importante solução para os casos onde uma rede de comunicação é utilizada, dado que as previsões podem ser exploradas para manter o AGV na trajetória correta em situações de atraso ou perda de pacotes na rede. Com base em experimentos realizados anteriormente com o Controle Preditivo Baseado em Modelo (Omena et al., 2021) e na revisão bibliográfica realizada, a arquitetura proposta se apresenta como uma solução efetiva para a navegação de AGVs nas fábricas inteligentes da Indústria 4.0, inclusive quando houver degradação do sinal na rede.

Uma importante ferramenta adotada para a implementação da arquitetura proposta é o ROS <sup>1</sup> (*Robot Operating System*). O ROS é um sistema operacional de robôs, em que se permite que diversas funções de um sistema robótico sejam executadas em diferentes nós. Por exemplo, o con-

trolador e o sistema de localização podem ser executados em diferentes nós, com troca de mensagens realizada por tópicos através do método publicar/subscrever. Também funciona em rede, ou seja, na arquitetura apresentada é possível trocar mensagens entre a borda e os AGVs. Outro recurso importante do ROS é a execução de serviços. Na proposta apresentada o controle preditivo pode ser executado como um serviço, assim, sempre que for necessário conduzir um AGV de um ponto a outro o serviço pode ser solicitado.

As próximas seções estão dispostas conforme a seguir. A Seção 2 contém uma breve teoria sobre o Controle Preditivo Baseado em Modelo. Uma discussão sobre experimentos realizados anteriormente com o controle preditivo e a computação na borda está apresentada na Seção 3. Na Seção 4 está a revisão bibliográfica. A arquitetura proposta é apresentada na Seção 5. Por fim, na Seção 6 estão as conclusões.

## 2. O CONTROLE PREDITIVO BASEADO EM MODELO

Uma breve teoria extraída do livro de Camacho and Bordons (2007) sobre o Controle Preditivo Baseado em Modelo está apresentada a seguir.

O Controle Preditivo Baseado em Modelo, na língua inglesa chamado de *Model Based Predictive Control* ou *Model Predictive Control* (a sigla associada às iniciais do segundo termo é utilizada aqui), teve início no final da década de 1970 e passou por consideráveis aprimoramentos desde então. O MPC não se refere a uma estratégia de controle específica, mas a uma ampla variedade de métodos de controle que fazem o uso explícito do modelo do processo para obter o sinal de controle através da minimização de uma função custo. As ideias gerais, que podem aparecer em maior ou menor grau na família do controle preditivo, são: (1) uso explícito do modelo para prever a saída do processo nos próximos instantes de tempo (horizonte); (2) cálculo de uma sequência de controle que minimize uma função custo; (3) estratégia retrocedente, portanto, em cada instante o horizonte é deslocado para o futuro, e o primeiro sinal de controle da sequência calculada em cada iteração é aplicado.

Algumas das vantagens do MPC em relação aos outros métodos de controle, são: (1) é atrativo para os que têm conhecimento limitado de controle, visto que os conceitos são intuitivos e ao mesmo tempo a sintonização é relativamente fácil; (2) pode ser utilizado para controlar uma variedade de processos, desde os que tenham uma dinâmica simples até os mais complexos e pode lidar facilmente com múltiplas variáveis; (3) tem compensação intrínseca para tempo morto; (4) introduz o controle *feedforward* de maneira natural para compensar os distúrbios; (5) o tratamento das restrições é simples e (6) é útil quando as referências futuras (robótica ou processos em lote) são conhecidas. Uma das desvantagens é a necessidade de mais poder computacional em relação a outros controladores, principalmente quando restrições (relacionadas às limitações de estado do processo e limites das entradas de controle) são consideradas. Entretanto, com o poder computacional disponível hoje isso não seria um grande problema. Outra desvantagem é a dependência do modelo

<sup>1</sup> <https://www.ros.org/>

do processo, dado que os benefícios obtidos com o uso do MPC serão afetados pela discrepância entre o processo real e o modelo utilizado.

Na implementação do MPC um modelo é utilizado para prever as saídas futuras da planta, baseado nas próximas ações de controle otimizadas. Essas ações são calculadas pelo otimizador levando em consideração a função custo (onde os erros futuros de rastreamento são considerados) como também as restrições.

### 3. EXPERIMENTOS ANTERIORES COM O MPC NA BORDA

Foram realizados experimentos com o MPC executado na borda para o controle de um robô móvel, com atrasos e perdas de pacotes em uma rede de comunicação, cujos resultados foram publicados em Omena et al. (2021). Em linhas gerais, os experimentos foram realizados para avaliar se os sinais da sequência de controle gerados pelo MPC no lado borda, podem, ser usados no lado local para manter o robô na trajetória, de forma estável, quando houver atrasos e perdas de pacotes na rede e um novo sinal de controle não seja recebido.

A arquitetura foi dividida entre o lado da borda e o lado local (ou do robô). Uma máquina virtual foi utilizada para representar a borda e executar o código Python com o MPC. O sistema operacional hospedeiro foi utilizado para execução do código Python local e o simulador de robôs CoppeliaSim<sup>2</sup>. O código local tem as rotinas do sensor para amostrar o estado do robô e enviar para a borda, como também as rotinas do atuador e do compensador, para aplicar os sinais de controle e realizar as compensações quando necessário. O compensador permanece aplicando os sinais da última sequência de controle recebida quando houver atrasos ou perdas de pacotes na rede. O modelo de robô utilizado no CoppeliaSim foi o Pioneer 3-DX, tipo de robô de tração diferencial. Diferente da arquitetura proposta na Seção 5, o controlador de baixo nível não foi aplicado nos experimentos.

Dois casos foram estudados: o da estabilização em um ponto e o do rastreamento da trajetória. Na estabilização em um ponto, o MPC recebe apenas o valor da referência final, que, no caso do robô móvel, são as coordenadas e o ângulo de orientação. Portanto, a solução do problema de otimização do MPC deve gerar uma trajetória, limitada ao horizonte de predição, juntamente com os respectivos sinais de controle para conduzir o robô até a referência. No rastreamento da trajetória, o MPC recebe um vetor com  $N$  (tamanho do horizonte de predição) referências, contendo as posições que o robô deve estar nos próximos  $N$  instantes, espaçados no tempo por  $T$  (intervalo das predições).

A partir de uma rede virtual entre os dois sistemas operacionais, diferentes perfis de rede foram emulados na conexão entre o lado da borda e o lado local, alguns utilizando o emulador de rede do Linux, o NetEm<sup>3</sup>, e outros utilizando o ERRANT (Trevisan et al., 2020). O último é uma ferramenta que se baseia em dados de medições obtidos em redes de comunicações reais para emular diferentes tipos de

rede. Os perfis '5G' e '200msPL50%' foram configurados no NetEm, o primeiro com uma latência de 1 ms e o segundo com uma latência de 200 ms e perdas de pacotes numa taxa de 50%. Os parâmetros desse segundo perfil foram escolhidos mais altos para de fato avaliar o desempenho da abordagem de compensação diante de atrasos e perdas de pacotes mais extremos. Os perfis 3G e 4G foram emulados no ERRANT. Dado que a ferramenta emula a rede baseada em dados de medições reais, existem diferentes situações de largura de banda e latência para uma única tecnologia de rede, as quais podem estar relacionadas à qualidade do sinal, por exemplo. Assim, as condições da rede nos perfis 3G e 4G foram configuradas no emulador para serem alteradas a cada 5 segundos.

As trajetórias obtidas nos experimentos estão apresentadas na Figura 1. Nos experimentos de estabilização em um ponto (Figura 1a), verifica-se que as trajetórias do robô são similares em todos os perfis de rede, exceto no perfil '200msPL50%', com trajetória ligeiramente diferente. O círculo vermelho delimita a região com o estado inicial, enquanto o verde é a região da referência ou estado final. O círculo azul delimita a região onde o robô faz uma manobra antes de estabilizar, visto que o ângulo de orientação da referência é 180 graus. Nesse caso é necessário girar o robô no sentido anti-horário e depois aplicar uma velocidade linear reversa.

Os resultados são mais satisfatórios nos experimentos de rastreamento da trajetória. Nessa modalidade o algoritmo do MPC leva em consideração uma referência que varia no tempo, com todos os pontos de trajetória que o robô irá percorrer nos instantes futuros. Assim, os sinais de controle são adequadamente ajustados e a trajetória do robô torna-se menos suscetível às oscilações da rede. Duas trajetórias foram utilizadas, a trajetória circular (Figura 1b) e a trajetória na forma do número oito (Figura 1c), com resultados bastante similares de trajetória em todos os perfis de rede, inclusive no perfil com maior atraso e perdas de pacote. Nos dois casos, os estados iniciais e finais são delimitados pelos círculos verde e vermelho, respectivamente, de modo que o robô completa a primeira volta e percorre o trecho inicial da segunda.

### 4. REVISÃO BIBLIOGRÁFICA

O controle e a coordenação de AGVs passa por diversas etapas. De Ryck et al. (2020) consideram que um sistema completo de gerenciamento de AGVs pode ser decomposto em cinco atividades: (1) alocação de tarefas; (2) planejamento da rota; (3) localização; (4) planejamento do movimento; (5) gerenciamento dos veículos. O foco deste trabalho é dar suporte da navegação dos AGVs, portanto, apenas as atividades de planejamento da rota e do movimento são consideradas.

Madridano et al. (2021) distribuem os algoritmos de planejamento da rota em quatro categorias: baseados em decomposição de grafos, em amostragem, em modelos matemáticos e bioinspirados. Nos métodos de decomposição de grafos alguns exemplos de algoritmos são o Dijkstra, A\* e D\*, enquanto nos métodos baseados em amostragem existem o Mapa de Rotas Probabilístico, Diagrama de Voronoi, Árvore Aleatória de Exploração Rápida e Campo

<sup>2</sup> <https://www.coppeliarobotics.com/>

<sup>3</sup> Network Emulator: <https://wiki.linuxfoundation.org/networking/netem>

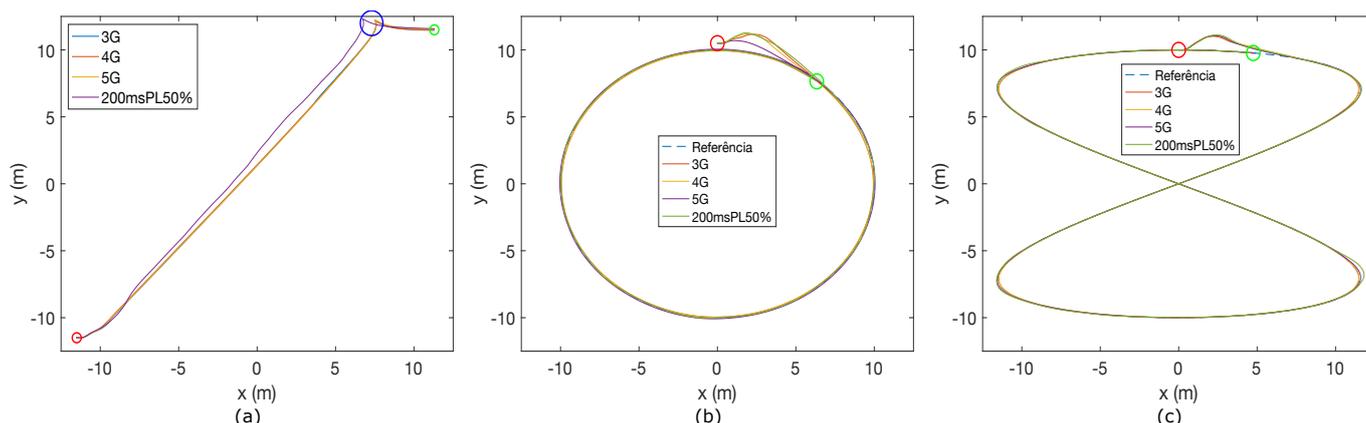


Figura 1. Trajetórias obtidas nos experimentos anteriores com execução do MPC na borda e um compensador no lado local: (a) estabilização em um ponto, (b) e (c) rastreamento da trajetória com referência circular e na forma do número oito, respectivamente.

Potencial Artificial. O MPC é incluído nos métodos baseados em modelos matemáticos, além da Programação Linear Interira Mista e da Programação Quadrática Inteira Mista. Nos métodos bioinspirados se encontram as Redes Neurais, Algoritmo Genético, Otimização por Enxame de Partículas, dentre outros.

Enquanto segue a rota planejada, o AGV pode se deparar com obstáculos, sejam eles obstáculos estáticos que não foram considerados pelo planejador, ou mesmo pessoas e outros AGVs que estejam navegando. Outra situação que pode ocorrer é o *deadlock*. Isso acontece quando o AGV não tem ação e não se move para frente ou para trás. Portanto, a modificação da rota planejada para evitar colisões e *deadlocks* é denominada planejamento do movimento (De Ryck et al., 2020).

Na navegação de robôs móveis ou AGVs o termo “Planejamento da Trajetória” também é comumente utilizado. O planejamento da rota tem como resultado uma rota que é independente do tempo, ou seja, um conjunto de pontos que formam a rota são distribuídos no espaço, mas sem associação com o tempo de alcance de cada um dos pontos (Bo et al., 2017). O planejamento da trajetória tem relação com o tempo, portanto, estima-se o tempo que o robô deve alcançar cada ponto da trajetória planejada.

Mercy et al. (2018) e de Angelis et al. (2019) dividem o planejamento da trajetória em abordagens acopladas e desacopladas. Um planejamento da trajetória acoplado soluciona um problema de otimização, tendo como resultado a trajetória otimizada e os correspondentes sinais de controle para conduzir o veículo ao longo da trajetória. A maioria dos métodos aplicam a abordagem desacoplada, dividindo o problema em planejamento da rota e em seguida o rastreamento da rota. No rastreamento o MPC é comumente utilizado para conduzir o veículo na rota planejada, visto que esta técnica de controle pode levar em consideração as limitações do sistema. Embora as abordagens desacopladas geralmente tenham problemas mais fáceis de resolver em relação às acopladas, elas retornam resultados subótimos, além de que podem ser impraticáveis pelo fato de que a cinemática dos veículos e as suas limitações não estão incluídas no problema de planejamento da rota. Portanto, os autores utilizam um algoritmo de controle ótimo para

gerar a trajetória que conduz o veículo em um ambiente com obstáculos móveis. Junto com a trajetória, a solução do problema também retorna os sinais de controle para condução do veículo. Seguindo o mesmo princípio, Mendes Filho et al. (2017) apresentam um problema de controle ótimo para planejamento da trajetória, incluindo as restrições cinemáticas e dinâmicas dos robôs. Os autores implementam um planejamento do movimento similar a um MPC clássico, mas diferente pelo fato de que não apenas o primeiro valor da solução otimizada é aplicado na entrada do sistema. As trajetórias geradas incluem a cinemática do robô, mas não a dinâmica. Numa segunda etapa, os autores então utilizam o MPC para rastrear os robôs nas trajetórias planejadas, incluindo agora as restrições dinâmicas.

A execução dos sistemas de controle de AGVs podem ocorrer de forma centralizada ou descentralizada. No primeiro caso, um único computador controla toda a frota de AGVs, implementando otimizações a partir das informações globais. A unidade central tem acesso às posições dos veículos e às informações das tarefas a serem executadas. Estes dados são usados pela entidade central para que as atividades sejam alocadas aos robôs e o caminho para conclusão das tarefas seja planejado de forma otimizada, evitando colisões com outros AGVs e *deadlocks* (De Ryck et al., 2020). No controle descentralizado, a decisão é distribuída aos AGVs para que operem de modo independente na busca de alcançar os objetivos globais (De Ryck et al., 2020).

Oyekanlu et al. (2020) abordam os potenciais benefícios ao utilizar a *Multi-Access Edge Computing* e a rede 5G, seja para um controle centralizado ou descentralizado. No modo centralizado, a central de controle mantém o estado de toda a frota e determina a rota que cada AGV deve percorrer, evitando colisões ou *deadlocks*. No modo descentralizado, os AGVs se comunicam e arbitram as tarefas entre si utilizando as antenas distribuídas da rede 5G. Com a MEC e a rede 5G, os AGVs se comunicam e trocam dados com o alocador de tarefas, estações de trabalho e outras máquinas para minimizar o tempo de transporte ao utilizar o AGV mais próximo para executar uma tarefa. Em situações de *deadlock*, os AGVs trocam dados entre si para solucionar o problema caso o controle

seja descentralizado. Se centralizado, os AGVs trocam dados com o controlador central para solucionar o conflito.

Com a popularização da robótica e a expansão das áreas de aplicação, as tarefas a serem executadas pelos robôs estão tornando-se mais complexas, o que vai na oposição de seus recursos computacionais limitados. O conceito da robótica em nuvem tira dos robôs a necessidade de terem grande capacidade computacional e memória, mas os conectam à nuvem para execução de serviços sempre que necessário, além de reduzir seus custos (Xu and Bian, 2020). Algumas aplicações utilizando a nuvem são apresentadas por Cardarelli et al. (2017), com uma arquitetura de robótica em nuvem que dão suporte na coordenação de AGVs, e também por Lopez et al. (2017), onde o MPC utilizado para planejar a trajetória é processado no próprio robô, mas com uso da nuvem para o compartilhamento de trajetórias, que, por fim, são utilizadas no procedimento de prevenção de colisões.

A combinação de computação na borda com redes móveis é explorada por Lambrecht et al. (2019), onde os autores apresentam um estudo de caso que utiliza uma rede 4G junto com a computação na borda, para o controle de sistemas de transporte autônomos em um ambiente industrial. O SLAM (*Simultaneous Localization and Mapping*), técnica utilizada para localizar o robô e mapear o ambiente, junto com o planejamento de trajetória, são transferidos para a borda, enquanto o robô executa localmente apenas funções básicas de segurança. Considerando as degradações no sinal de rádio que podem acontecer no ambiente industrial, Nakimuli et al. (2021) avaliam o desempenho do controle de um AGV com computação na borda e rede 5G, na ocorrência de atrasos e perdas de pacotes.

Nos casos em que o controlador e a planta estão conectados por uma rede de comunicação, a adoção do MPC como controlador para compensar o atraso ou perda de pacotes em redes de comunicação já é uma área explorada. Por se tratar de uma estratégia de controle que realiza previsões dentro de um horizonte finito, o MPC pode ser utilizado para minimizar tais efeitos. A cada estado amostrado, a sequência de sinais de controle gerada pelo MPC é enviada à planta. Durante a operação normal, apenas o primeiro sinal é aplicado e o restante é descartado. Em caso de atrasos ou perdas de pacotes, os sinais de controle dentro da última sequência recebida são utilizados até que um novo pacote com uma sequência mais recente seja recebido (Liu et al., 2006; Grüne et al., 2009).

Trabalhos conduzidos na Suécia pela Universidade de Lund, em conjunto com a empresa da área de telecomunicações, Ericsson, mesclam o uso do MPC com computação na borda e redes 5G. Skarin et al. (2018) avaliam o desempenho do MPC como controlador de um processo de equilíbrio de uma esfera em uma viga que se movimenta em torno de um eixo, implementado em nós com diferentes capacidades de processamento, mas também hospedados em locais diferentes. Em outro trabalho, Skarin et al. (2019) implementam o MPC na borda, porém, um controlador local também é aplicado. No modo assistido, o MPC executado na nuvem ou na borda, é responsável por controlar o processo da esfera e viga. Se os pacotes estiverem atrasados ou forem perdidos, o sistema é mudado para o modo local e é controlado por um controlador mais

simples, o Regulador Linear Quadrático (LQR – *Linear Quadratic Regulator*).

#### 4.1 Discussão

Dos algoritmos citados nesta revisão bibliográfica, com exceção dos baseados em modelos matemáticos, as outras possibilidades não consideram as restrições cinemáticas e dinâmicas do robô (Madridano et al., 2021), o que pode resultar em rotas impraticáveis a serem seguidas, dado que as limitações do veículo não são consideradas na fase de planejamento da rota (Mercy et al., 2018). As técnicas de controle ótimo, por sua vez, além de incluir as restrições cinemáticas e dinâmicas, funcionam como uma abordagem acoplada, além de que o problema das colisões pode ser incluído através de restrições no processo de otimização (Mercy et al., 2018; de Angelis et al., 2019). Mendes Filho et al. (2017), por exemplo, além de utilizarem o controle ótimo para planejamento da trajetória livre de colisões, incluindo os modelos cinemáticos e dinâmicos, utilizam o MPC para rastrear os robôs na trajetória planejada na etapa anterior.

Quando os sistemas de controle de robôs móveis são assistidos por computação em nuvem ou na borda, Cardarelli et al. (2017) apresentam uma arquitetura de robótica em nuvem que utiliza dados globais para planejamento da navegação, entretanto, os AGVs são limitados a navegarem em rotas fixas. Lopez et al. (2017) propõem uma arquitetura em nuvem com a prevenção de colisões resolvida localmente entre os robôs, com o compartilhamento das trajetórias previstas realizada na nuvem, o que pode causar colisões em caso de problemas na comunicação. Nakimuli et al. (2021) avaliam o controle do AGV com computação na borda e rede 5G, sob diversas condições de atrasos e perdas de pacotes, entretanto, apenas analisam o desempenho do controle e não aplicam nenhuma estratégia para tratar dos problemas decorrentes dos efeitos da rede. Lambrecht and Funk (2020) fazem a transferência de atividades de planejamento e SLAM para a borda, utilizando uma rede 4G, mas apenas avaliam o desempenho do controle e a redução do consumo de energia no robô quando o processamento das atividades é transferido para a borda, além de que não utilizam múltiplos robôs.

Como referências para a construção da arquitetura apresentada a seguir, têm-se as propostas em Mercy et al. (2018) e de Angelis et al. (2019), na adoção do MPC para planejamento da trajetória com prevenção de colisões, como também a proposta de Lopez et al. (2017), na adoção de um planejador global na nuvem e o MPC para o planejamento da trajetória livre de colisões. Nesse último trabalho, o MPC é executado no robô, o que difere da proposta apresentada aqui, onde o planejamento da trajetória e a prevenção de colisões já é resolvida globalmente na borda, com o rastreamento realizado no robô. A estratégia adotada tem uma certa similaridade com a abordagem de Mendes Filho et al. (2017), porém, com a adição da computação na borda para o planejamento.

## 5. ARQUITETURA DE CONTROLE DE AGVS COM COMPUTAÇÃO NA BORDA

Baseado na discussão anterior e nos resultados apresentados em Omena et al. (2021), é proposta neste trabalho uma

arquitetura para controle de AGVs com aplicação do MPC em duas camadas, sendo uma na borda e outra nos AGVs, de tal forma que o sistema esteja apto a lidar com situações em que a comunicação entre a borda e o AGV esteja sujeita a degradação no sinal da rede. O sistema é centralizado no sentido de que centraliza-se na borda o planejamento da trajetória a partir de informações globais, mas ao mesmo tempo, assume-se uma postura descentralizada no lado local (ou do AGV) quando o veículo deve seguir a trajetória planejada, ou até mesmo replanejá-la quando ocorrer a degradação no sinal da rede. Neste caso, assim que o sinal for restabelecido, a borda toma conhecimento da posição do AGV e volta a assumir o planejamento. A centralização do controle na borda viabiliza o acesso às informações globais da frota de AGVs. Assim, as informações de estado dos AGVs e do ambiente são conhecidas, auxiliando a execução do algoritmo do planejamento da trajetória na escolha da que seja mais rápida e livre de colisões. Pode auxiliar também na escolha do AGV que esteja mais apto a realizar uma determinada tarefa, alocando-a, por exemplo, ao veículo que esteja mais próximo, que tenha capacidade para transportar a carga requerida e que o estado da bateria seja suficiente para conclusão da tarefa. No mais, enquanto centralizando na borda, o sistema de controle de navegação dos AGVs estará integrado ao sistema de gerenciamento de produção da fábrica.

Além de fazer uso dos recursos computacionais da borda, os AGVs trafegam em rotas livres, sem a necessidade de percorrer caminhos fixados por linhas ópticas ou indutivas, por exemplo, como também não dependem de sensores estacionários. Essa abordagem permite que o sistema se adapte facilmente às alterações de layout, com um custo mais baixo, considerando que não será necessário realizar mudanças físicas na infraestrutura da fábrica.

Na Figura 2 está apresentada a arquitetura proposta. As regiões tracejadas representam diferentes sistemas computacionais. No lado esquerdo está a representação da computação na borda com a execução de  $n$  blocos de MPC para planejamento da trajetória (1), para cada um dos  $n$  AGVs utilizados. Conforme discutido anteriormente, o MPC pode planejar a trajetória levando em consideração as restrições cinemáticas, prevenção de colisões, restrições de estado e restrições dos sinais de controle, portanto, esse método foi escolhido nessa etapa. Os blocos de MPC planejam as trajetórias de seus respectivos AGVs e trocam entre si os dados de trajetórias para que cada um possa utilizar no procedimento de prevenção de colisões. Os dados de trajetórias são enviados também aos AGVs, nos quais respondem com os seus estados.

O MPC para planejamento da trajetória funciona conforme o fluxograma da Figura 3. Após inicializado, se o AGV já estiver na referência desejada, o processo é finalizado. Caso contrário, a trajetória é calculada e enviada ao AGV. No próximo instante de cálculo, definido pelo período  $T$ , se o estado do AGV foi recebido, o processo é reiniciado. Se não, estima-se que o AGV está no próximo ponto da última trajetória calculada. O processo é então executado até que o AGV alcance a referência. A periodicidade  $T$  de execução do MPC é o mesmo intervalo de tempo entre os pontos das trajetórias previstas geradas.

Uma outra maneira de se planejar a trajetória é utilizar um único MPC para todos os AGVs. Esse método intrinsecamente evita os *deadlocks*, entretanto, o problema de otimização aumenta junto com a quantidade de AGVs (Salimi Lafmejani and Berman, 2021). Dessa forma, é proposta na arquitetura a utilização de diferentes processos de MPC, um para cada robô, executados com o ROS, por onde se faz o compartilhamento de trajetórias através de tópicos. Um problema de otimização é então subdividido em problemas menores e resolvidos de forma paralela.

No lado direito da Figura 2 estão representados os sistemas computacionais embarcados nos AGVs. Considera-se aqui que os computadores embarcados nos AGVs são de baixo custo, com restrições nos recursos computacionais, mas com capacidade suficiente para processamento em tempo hábil dos algoritmos necessários. Cada um dos AGVs recebe a respectiva trajetória, a ser utilizada pelo bloco do MPC para rastreamento da trajetória (2). O rastreamento da trajetória utilizado aqui é o mesmo do apresentado em Omena et al. (2021), portanto, todos os  $N$  pontos de trajetória dentro do horizonte de predição são utilizados para o cálculo do sinal de controle referente às velocidades linear e angular. Essa abordagem foi escolhida com base nos resultados daquele trabalho, considerando que nas situações de degradação do sinal da rede, é possível manter o AGV na trajetória previamente planejada. Por fim, o sinal de controle gerado é enviado ao controlador de baixo nível (3) para que as velocidades linear e angular sejam devidamente aplicadas ao chassi do AGV.

Vale notar que os blocos de MPC executados na borda e no AGV processam problemas de otimização de diferentes complexidades. No planejamento da trajetória na borda, os blocos de MPC incluem o procedimento para prevenção de colisões entre os AGVs e os obstáculos fixos. O compartilhamento dos dados de trajetórias é feito ainda na borda, portanto, não estará sujeito aos atrasos ou perdas na rede. Os AGVs executam o rastreamento da trajetória apenas para o respectivo veículo, solucionando um problema mais simples, com menor consumo de energia, que, como consequência, aumentará a autonomia da bateria.

Nas situações em que a trajetória não seja atualizada por conta de atrasos ou perdas dos pacotes na rede e caso a última trajetória recebida da borda seja totalmente utilizada, o AGV pode planejar sua trajetória de forma independente e descentralizada com um algoritmo mais simples, onde os obstáculos estáticos e dinâmicos (incluindo outros AGVs) no seu entorno são considerados. Os obstáculos podem ser detectados por sensores a laser ou câmeras, por exemplo. A arquitetura simplificada que representa esta situação para apenas um AGV está apresentada na Figura 4. A trajetória planejada (1) é repassada também ao MPC para rastreamento da trajetória (2), que por sua vez repassa os sinais de controle para o controlador de baixo nível (3).

## 6. CONCLUSÃO

A partir da revisão bibliográfica apresentada na Seção 4, foi verificado o potencial do MPC para o controle de robôs móveis, considerando a sua ação preditiva e a otimização da trajetória em relação ao tamanho do percurso ou outros fatores. O MPC permite de forma fácil a inclusão das

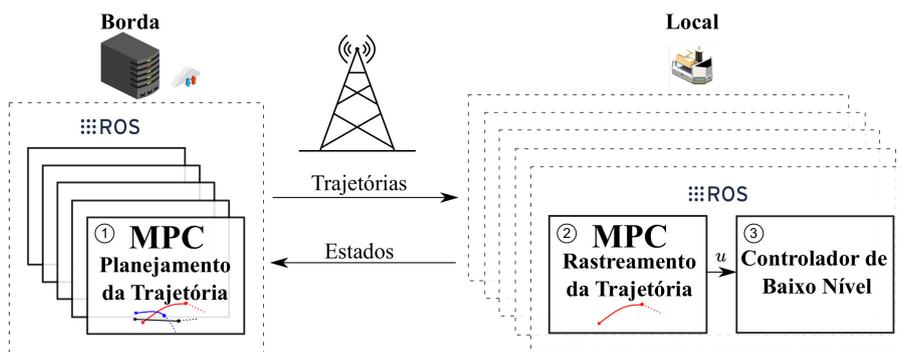


Figura 2. Arquitetura proposta. As atividades executadas no lado da borda estão à esquerda e no lado local ou do AGV estão à direita.

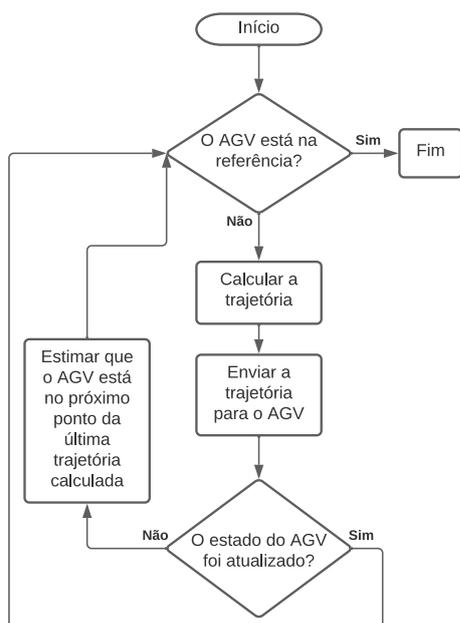


Figura 3. Fluxograma do MPC para planejamento da trajetória.

restrições cinemáticas, dinâmicas e de prevenção de colisões no problema de otimização. Observou-se que a ação preditiva do MPC é pouco explorada para lidar com os problemas relacionados ao uso das redes de comunicação, tais como os atrasos e perdas de pacotes, num cenário de transferência do controlador para a nuvem ou borda, principalmente em aplicações com robôs móveis, ou mais especificamente, com AGVs.

Ainda na revisão bibliográfica foi observada a necessidade de manter uma camada de controle no lado local, de forma a reduzir oscilações no sinal de controle, assegurar a esta-

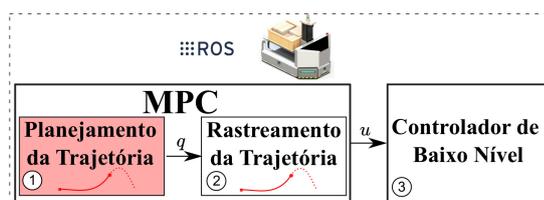


Figura 4. Arquitetura simplificada, utilizada em situações em que o AGV deve navegar sem o suporte da borda.

bilidade do robô e que a trajetória planejada pela borda seja respeitada. Além disso, considerando a utilização de múltiplos robôs, é necessário assegurar que não haverá colisões entre os AGVs, ou entre os AGVs e pessoas ou objetos. Os resultados dos experimentos anteriores e a revisão bibliográfica foram então a base para a construção da arquitetura detalhada na Seção 5. Pode-se dizer que o MPC aplicado na borda funciona como um controlador de alto nível, planejando a trajetória de todos os AGVs a partir do conhecimento dos dados globais, e que a camada de controle no lado local deve respeitar o planejamento global ou refazer o planejamento caso seja necessário.

Mesmo que as redes de nova geração sejam utilizadas, o ambiente industrial está sujeito a diferentes tipos de interferências, o que pode comprometer o controle de AGVs com sistema de computação na borda e redes sem fio. Dessa forma, a arquitetura proposta, na qual mescla o MPC com computação na borda, se apresenta como uma importante solução de controle de AGVs na indústria.

#### AGRADECIMENTOS

Os autores agradecem ao Programa de Pós-Graduação em Engenharia Elétrica (COPELE) e ao VIRTUS - Núcleo de Pesquisa, Desenvolvimento e Inovação em Tecnologia da Informação, Comunicação e Automação, da Universidade Federal de Campina Grande.

#### REFERÊNCIAS

- Aazam, M., Zeadally, S., and Harras, K.A. (2018). Deploying fog computing in industrial internet of things and industry 4.0. *IEEE Transactions on Industrial Informatics*, 14(10), 4674–4682.
- Al-Fuqaha, A., Guizani, M., Mohammadi, M., Aledhari, M., and Ayyash, M. (2015). Internet of things: A survey on enabling technologies, protocols, and applications. *IEEE Communications Surveys Tutorials*, 17(4), 2347–2376.
- Bo, N., Li, X., Dai, J., and Tang, J. (2017). A hierarchical optimization strategy of trajectory planning for multi-uavs. In *2017 9th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC)*, volume 1, 294–298.
- Camacho, E.F. and Bordons, C. (2007). *Introduction to Model Predictive Control*, 1–11. Springer London, London.

- Cardarelli, E., Digani, V., Sabattini, L., Secchi, C., and Fantuzzi, C. (2017). Cooperative cloud robotics architecture for the coordination of multi-agv systems in industrial warehouses. *Mechatronics*, 45, 1–13.
- de Angelis, E.L., Giulietti, F., Pipeleers, G., Rossetti, G., and Van Parys, R. (2019). Optimal autonomous multi-robot motion planning in an obstructed environment. *Aerospace Science and Technology*, 87, 379–388.
- De Ryck, M., Versteyhe, M., and Debrouwere, F. (2020). Automated guided vehicle systems, state-of-the-art control algorithms and techniques. *Journal of Manufacturing Systems*, 54, 152 – 173.
- García-Morales, J., Lucas-Estañ, M.C., and Gozalvez, J. (2019). Latency-sensitive 5g ran slicing for industry 4.0. *IEEE Access*, 7, 143139–143159.
- Grüne, L., Pannek, J., and Worthmann, K. (2009). A prediction based control scheme for networked systems with delays and packet dropouts. In *Proceedings of the 48th IEEE Conference on Decision and Control (CDC) held jointly with 2009 28th Chinese Control Conference*, 537–542.
- Gubbi, J., Buyya, R., Marusic, S., and Palaniswami, M. (2013). Internet of things (iot): A vision, architectural elements, and future directions. *Future Generation Computer Systems*, 29(7), 1645 – 1660.
- Kiani, A. and Ansari, N. (2018). Edge computing aware noma for 5g networks. *IEEE Internet of Things Journal*, 5(2), 1299–1306.
- Lambrech, J. and Funk, E. (2020). Edge-enabled autonomous navigation and computer vision as a service: A study on mobile robot’s onboard energy consumption and computing requirements. In M.F. Silva, J. Luís Lima, L.P. Reis, A. Sanfeliu, and D. Tardioli (eds.), *Robot 2019: Fourth Iberian Robotics Conference*, 291–302. Springer International Publishing, Cham.
- Lambrech, J., Steffens, E.J., Geitz, M., Vick, A., Funk, E., and Steigerwald, W. (2019). Cognitive edge for factory: a case study on campus networks enabling smart intralogistics. In *2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1325–1328.
- Liu, G.P., Mu, J.X., Rees, D., and Chai, S.C. (2006). Design and stability analysis of networked control systems with random communication time delay using the modified mpc. *International Journal of Control*, 79(4), 288–297.
- Lopez, F.G., Abbenseth, J., Henkel, C., and Dörr, S. (2017). A predictive online path planning and optimization approach for cooperative mobile service robot navigation in industrial applications. In *2017 European Conference on Mobile Robots (ECMR)*, 1–6.
- Madridano, A., Al-Kaff, A., Martín, D., and de la Escalera, A. (2021). Trajectory planning for multi-robot systems: Methods and applications. *Expert Systems with Applications*, 173, 114660.
- Mao, Y., You, C., Zhang, J., Huang, K., and Letaief, K.B. (2017). A survey on mobile edge computing: The communication perspective. *IEEE Communications Surveys & Tutorials*, 19(4), 2322–2358. doi:10.1109/COMST.2017.2745201.
- Mendes Filho, J.M., Lucet, E., and Filliat, D. (2017). Real-time distributed receding horizon motion planning and control for mobile multi-robot dynamic systems. In *2017 IEEE International Conference on Robotics and Automation (ICRA)*, 657–663.
- Mercy, T., Van Parys, R., and Pipeleers, G. (2018). Spline-based motion planning for autonomous guided vehicles in a dynamic environment. *IEEE Transactions on Control Systems Technology*, 26(6), 2182–2189.
- Nakimuli, W., Garcia-Reinoso, J., Sierra-Garcia, J.E., Serrano, P., and Fernández, I.Q. (2021). Deployment and evaluation of an industry 4.0 use case over 5g. *IEEE Communications Magazine*, 59(7), 14–20.
- Omena, R., Santos, D., and Perkusich, A. (2021). An approach to reduce network effects in an industrial control and edge computing scenario. In *Proceedings of the 11th International Conference on Cloud Computing and Services Science - CLOSER*, 296–303. INSTICC, SciTePress.
- Oyekanlu, E.A., Smith, A.C., Thomas, W.P., Mulroy, G., Hitesh, D., Ramsey, M., Kuhn, D.J., Mcghinnis, J.D., Buonavita, S.C., Looper, N.A., Ng, M., Ng’oma, A., Liu, W., McBride, P.G., Shultz, M.G., Cerasi, C., and Sun, D. (2020). A review of recent advances in automated guided vehicle technologies: Integration challenges and research areas for 5g-based smart manufacturing applications. *IEEE Access*, 8, 202312–202353.
- Qiu, T., Chi, J., Zhou, X., Ning, Z., Atiquzzaman, M., and Wu, D.O. (2020). Edge computing in industrial internet of things: Architecture, advances and challenges. *IEEE Communications Surveys Tutorials*, 22(4), 2462–2488.
- Salimi Lafmejani, A. and Berman, S. (2021). Nonlinear mpc for collision-free and deadlock-free navigation of multiple nonholonomic mobile robots. *Robotics and Autonomous Systems*, 141, 103774.
- Skarin, P., Eker, J., Kihl, M., and Årzen, K.E. (2019). Cloud-assisted model predictive control. In *2019 IEEE International Conference on Edge Computing (EDGE)*, 110–112.
- Skarin, P., Tärneberg, W., Årzen, K.E., and Kihl, M. (2018). Towards mission-critical control at the edge and over 5g. In *2018 IEEE International Conference on Edge Computing (EDGE)*, 50–57.
- Trevisan, M., Safari Khatouni, A., and Giordano, D. (2020). Errant: Realistic emulation of radio access networks. *Computer Networks*, 176, 107289.
- Xu, B. and Bian, J. (2020). A cloud robotic application platform design based on the microservices architecture. In *2020 International Conference on Control, Robotics and Intelligent System, CCRIS 2020*, 13–18.
- Zhao, P. and Dán, G. (2018). A benders decomposition approach for resilient placement of virtual process control functions in mobile edge clouds. *IEEE Transactions on Network and Service Management*, 15(4), 1460–1472.