

Uma Abordagem por Técnicas Adaptativas de Segmentos de Retas em Navegação Robótica

L. C. Barros Neto and A. H. Hirakawa

Abstract— Com enfoque em adaptatividade, foi investigada uma arquitetura de registro de dados que possibilite a construção de representações cartesianas abstratas de ambientes desconhecidos por robôs móveis. As tecnologias adaptativas são formalismos da ciência da computação capazes de alterar seu comportamento dinamicamente, sem a interferência de agentes externos, em resposta a estímulos de entrada. Assim, este trabalho introduz uma nova estrutura de dados, não cartesiana, que permite a representação do ambiente e do movimento do robô mais próximo do movimento natural esperado, com base em segmentos de linhas retas digitalizadas adaptativos (SLRDA).

Keywords— Adaptive Systems, Robotics, Automata, Computational Geometry, Pattern Recognition, Error Correction.

I. INTRODUÇÃO

CLASSICAMENTE, os agentes robóticos são normalmente equipados com sistemas de sensores e de atuadores, para percepção e atuação, respectivamente, no espaço físico ao qual estão integrados. Em robótica móvel, além dos sistemas de sensores e atuadores, os robôs são providos também de sistemas de deslocamento para que se movimentem pelo ambiente a fim de executar atividades de maior complexidade que as da robótica tradicional. Entende-se navegação como o processo ou atividade de planejamento de um caminho e posterior movimentação e direcionamento nesse caminho [1] para que um robô autônomo se desloque com segurança de um local para outro, sem se perder ou colidir com outros objetos, a fim de executar determinada tarefa. [2] afirma que em navegação robótica existem três problemas gerais que são localização de um objetivo ou alvo a alcançar, planejamento da rota ou caminho das partes móveis para atingir o objetivo, e controle do movimento na rota. Dentre estes três problemas, [2] argumenta que o planejamento de um caminho é uma das questões mais importantes no processo de navegação de robôs móveis autônomos, sendo uma área ativa de pesquisas desde a década de 1970.

Uma aproximação comumente utilizada em planejamento de trajetórias é considerar o ambiente como um espaço cartesiano a fim de facilitar a discriminação de configurações requeridas. No entanto, a implementação dos algoritmos respectivos em um espaço requer transformações de coordenadas cartesianas para conjuntos em tempo real, tendo em vista que o controle de movimentos do manipulador é feito prevendo as articulações mecânicas. Em tempo real significa

para [4] que a trajetória é implementada por um método com baixo custo computacional à medida que ocorre o movimento, próximo do ideal por não envolver atrasos ou escalas de tempos significativos que poderiam comprometer o controle da dinâmica do manipulador por seus atributos como tempo, posição, velocidade, aceleração. Portanto, a complexidade computacional envolvida na transformação de coordenadas é um dos aspectos importantes dos algoritmos de planejamento de trajetórias no espaço cartesiano para viabilizar aplicações em tempo real [3] [4].

Conforme [3], com base em especificações iniciais contendo um caminho intermediário e um caminho final, o planejamento de trajetórias leva em conta fatores como a posição articular das partes móveis, a velocidade e a aceleração de um manipulador robótico. Em [5] e [2] apresenta-se um resumo geral de técnicas e algoritmos, relacionadas ao controle do movimento e ao planejamento da trajetória de robôs, tais como: *i)* de esquema motor, *ii)* da velocidade do obstáculo, *iii)* método do cone de colisão, *iv)* abordagem por curvas polinomiais polares (baseadas em sensores), *v)* métodos de computação inteligentes tais como inteligência artificial (IA) ou redes neurais, *vi)* método de campo potencial artificial, e *vii)* método gráfico de visibilidade.

Independentemente de técnicas e algoritmos para controle do movimento e planejamento de trajetórias, sempre ocorrem interferências espúrias e aproximações que interferem com os modelos teóricos. [4] menciona que os trabalhos iniciais em planejamento de movimento de robôs concentraram-se principalmente em otimizações de tempos associadas a aproximações padronizadas deste movimento. As aproximações indicadas por [4] são de segmentos lineares combinados com curvas, por exemplo, arcos de forma parabólica, entre pontos de comutação selecionados da trajetória.

Comumente, o planejamento de trajetórias no espaço cartesiano costuma aproximar caminhos cartesianos por trajetórias compostas de segmentos de retas, desde que os erros de translação e rotação gerados pelo desvio entre um caminho cartesiano e as trajetórias correspondentes interpoladas possam ser ajustados para satisfazer tolerâncias especificadas. [3] restringe aproximação de caminhos cartesianos por segmentos de retas apenas para os que apresentem suavidade, ou seja, significando que um dado caminho cartesiano possa ser parametrizado por uma variável t de tempo e seja, pelo menos, derivável em segunda ordem com relação a t .

Ainda na opinião de [4], trabalhos posteriores estudaram requisitos mecânicos importantes, tais como limitações

L. C. de Barros Neto, Universidade de São Paulo (USP), São Paulo, Brasil, leonclarobr@gmail.com

A. H. Hirakawa, Universidade de São Paulo (USP), São Paulo, Brasil, arhiraka@usp.br

vibracionais de robôs em aplicações industriais, tendo em vista que uma trajetória suave, sem vibrações, é importante para reduzir o desgaste do manipulador robótico e melhorar a precisão e velocidade de rastreamento do mesmo. Limitação vibracional (e, proporcionalmente, da taxa de torque) resulta também em carga suavizada para os atuadores, o que efetivamente reduz as frequências ressonantes de excitação do manipulador, consequentemente minimizando desgastes do atuador. Além disso, tal requisito é importante para aplicações específicas, inclusive por questões de maior segurança.

A fim de se obter trajetórias mais suaves, a comutação de retas para arcos tem sido efetuada também por curvas, especialmente de terceiro grau, definidas matematicamente por dois ou mais pontos de controle (splines). Polinômios de ordem superior não são geralmente utilizados devido à sua tendência a oscilar, gerando movimentos retrógrados. Entretanto, [4] descreve um método que utiliza uma concatenação de polinômios de quinta ordem para produzir uma trajetória suave entre dois pontos do trajeto, aproximando segmentos lineares entre pontos de controle. A interligação destes pontos de controle com polinômios de quinto grau resulta em uma trajetória controlada que não oscila, considerada viável para utilização *on-line*. No que se refere às interpolações de quarto grau, o trabalho de [3] apresenta métodos para aproximar caminhos no espaço cartesiano, resultados do planejamento de trajetórias de robôs por conjunto de segmentos de retas, em que procedimentos sistemáticos são propostos por algoritmos recursivos de planejamento. A abordagem proposta é ilustrada por um exemplo numérico.

[6] examina o planejamento do caminho mais curto sob uma direção métrica euclidiana ponderada. Motivado pelo problema de encontrar as melhores rotas para veleiros, [6] mostra que o caminho mais curto sempre consiste em dois segmentos de retas, e por estas resolve-se o problema por figuras poligonais. Na prática de [6], o percurso de um veleiro é uma sequência finita de segmentos de retas, em que investiga a complexidade de várias instâncias do problema do marinheiro em um veleiro, incluindo o caso da trajetória ser uma poligonal. A seguir, a trajetória sempre consiste em segmentos de retas, e por estas resolve-se o problema por figuras poligonais. Finalmente, o percurso, composto por uma sequência finita de segmentos de retas, é compensado devido aos erros e aproximações.

Em determinadas aplicações críticas, os erros e aproximações de trajetórias por linhas retas ou por curvas com curvatura constante (arcos) têm que ser compensados nos modelos e implementações. Por exemplo, em [7] é apresentada uma abordagem baseada em trajetória de navegação para o controle lateral de veículos autônomos que percorrem uma fila e têm que seguir o caminho do carro líder por meio de um sistema de processamento de imagens sem utilizar qualquer infraestrutura adicional de comunicação ou de sistema de posicionamento global (GPS). A solução encontrada para que o veículo autônomo não se desvie da trajetória do veículo líder, devido aos erros e aproximações, foi um procedimento que envolve armazenar o histórico do posicionamento do

veículo líder e a velocidade do veículo autônomo com relação ao tempo.

Nesse estudo de [7], a transformação de coordenadas de posição em um sistema de coordenadas em repouso envolveu compensar o movimento do veículo a ser controlado usando a sua velocidade e seu ângulo da direção da trajetória. Conhecendo-se a posição do veículo a ser controlado em um sistema de coordenadas, pode-se selecionar o ponto de trajetória do veículo líder que está mais próximo da frente do veículo sob controle, como entrada para o controle lateral. No entanto, controladores laterais reais têm certo atraso para garantir a estabilidade e compensar erros de medição dos dados dos sensores. Para levar tais erros em consideração, o algoritmo sempre seleciona o ponto de trajetória que exceda certa distância mais adiante, medida a partir do veículo sendo controlado [7].

No aspecto de erros e aproximações de trajetórias por digitalizações, [8] apresenta um esquema em tempo discreto de planejamento de movimento para determinar os pontos definidores de trajetória em linha reta. A motivação de [8] é que os esforços anteriores em planejamento de trajetórias causam que o braço do manipulador pode não estar no caminho desejado, mas sobre um caminho interpolado por polinômios. Adicionalmente, devido a aproximações discretas de parâmetros como velocidade, aceleração e vibrações permitidas, tais soluções de otimização por meio de polinômios envolvem cálculos errôneos, sendo também computacionalmente intensivo, o que tende a inviabilizá-las em aplicações correntes. [8] opta por uma solução algorítmica formulando o problema de planejamento de trajetória como de maximização da distância cartesiana entre dois instantes no tempo com a restrição de que componham um determinado caminho em linha reta sujeito a limitações em torque e de suavidade da trajetória.

[5] pesquisa algoritmos de planejamento de trajetórias pelo método de campo potencial e método de Monte Carlo, para evitar obstáculos na navegação de robô móvel em um ambiente dinâmico do tipo industrial. O algoritmo de planejamento de trajetória é dividido em dois sub-módulos: o planejamento de trajetória global que usa grafos de visibilidade e o planejamento de caminho local pelo método de campo potencial para evitar os obstáculos. O processamento de imagens é usado para obter informações sobre o ambiente de trabalho a partir de câmera com acesso ao ambiente.

Tendo em vista que os erros no processo de digitalização envolvem também detalhes de calibração do sensor, por exemplo, uma câmera montada no robô, [9] argumenta que o planejamento no espaço euclidiano tende a resultar em trajetórias inadequadas no espaço da imagem e vice-versa. Estas dificuldades são devidas à transformação em perspectiva da câmera; à perda de uma dimensão, devido à projeção sobre o plano da imagem; e ao fato de que, em termos práticos, é considerada viável apenas uma aproximação grosseira de parâmetros da câmera. Assim, [9] propõe um esquema de planejamento de trajetórias por imagens tal que o comportamento em linha reta é assegurado tanto no espaço da

imagem quanto no espaço efetivo do mundo em que se movimenta o robô. Esse esquema permite trajetórias compactas em linhas retas, ou seja, para o centro óptico da câmera relativamente a um ponto escolhido arbitrariamente no plano da imagem.

Focado em técnicas sintáticas para compensar erros de calibração de sensores, o trabalho de [10] descreve uma abordagem de segmentação de imagens usando linhas retas, na verdade aproximações de arcos curvos por linhas retas codificadas por cadeias de símbolos ou *chain code*, com a finalidade de reduzir significativamente o número de primitivas, bem como diminuir o esforço computacional e a taxa de erros em segmentação. Os algoritmos propostos são parte de um sistema de computação de profundidade a partir de sequências de imagens monoculares tomadas de uma sequência de pontos de vista diferentes por uma câmera montada em um manipulador de robô. Tendo em vista que cada grupo de duas imagens consecutivas é considerado como uma imagem estéreo, o trabalho se propõe a reduzir os problemas de calibração imprecisa e intensidade variável de sinal recebida pela câmera visando fundamentar os trabalhos futuros em planejamento de trajetórias.

Sob o aspecto adaptativo para solução do problema de modelagem em navegação, [11] descreve evoluções de uma abordagem que parte da constatação de que um conjunto inicial de parâmetros de projeto ou de controle de um sistema muito frequentemente torna possível inferir o comportamento do sistema sob um cenário envolvendo conjunto de parâmetros diversos daquele inicial. Isto sugere fortemente o potencial para "aprendizado" sobre o comportamento global de um sistema de navegação, simplesmente partindo de um caminho considerado como amostra. Por sua vez, tal fato implica o potencial para ajustar vários parâmetros críticos, como um refinamento de modo a ajustar continuamente o desempenho do robô por reações a alterações e mudanças no ambiente. Assim, [11] é um estudo centrado em autômatos estocásticos para que o comportamento de um robô em navegação possa ser inferido a partir de um caminho único como amostra.

[12] enfatiza determinadas habilidades dos robôs autônomos tal como a capacidade de se adaptarem aos ambientes de operação, aprender conforme adquirem mais experiência e realizar escolhas frente às mudanças ou novas situações desses ambientes.

Concluindo, de modo geral a baixa fidelidade de representação utilizando métodos tradicionais com base em mapas ambientais cartesianos, resulta em pouco conhecimento armazenado sobre o ambiente ao redor do robô, conhecimento este comumente obtido por sensores de processamento de imagens associado a procedimentos de detecção de linhas retas e arcos, tal como em [13].

A hipótese inicial deste trabalho é que o formalismo de SLRDA apresentado em [14] e [15] seja um modelo mais preciso de representação do movimento e do ambiente em que está inserido o robô.

A. Organização do texto

A seção I analisa os trabalhos principais que embasam conceitualmente esta pesquisa. A seção II apresenta conceitos

necessários para o entendimento deste trabalho, tais como segmentos de linhas retas digitalizadas (SLRD), segmentos de linhas retas digitalizadas adaptativos (SLRDA) e autômato finito adaptativo (AFA). A seção III apresenta o enunciado do problema estudado por este trabalho, os desafios e os meios e métodos aplicados para superar os desafios. A seção IV apresenta a síntese do método proposto, com detalhes técnicos sobre o comentado no item anterior. A seção V especifica a implementação de um banco de autômatos que implementam SLRDA. A seção VI apresenta justificativas para o método de comparação nos experimentos e o posicionamento da pesquisa com relação ao estado da arte. A seção VII descreve os experimentos efetuados. A seção VIII relaciona perspectivas de trabalhos futuros. A seção IX apresenta as considerações finais deste relato e contribuições.

II. FUNDAMENTOS

Este tópico apresenta conceitos básicos necessários ao entendimento desta pesquisa por uma síntese de [15].

A. Conceitos básicos sobre SLRD

Em [16] o *chain code* foi apresentado como um descritor de contornos, de apenas um pixel de espessura, e um modelo definidor de retas digitais foi também conjecturado por Freeman. Nesse modelo, dado um ponto digital, as vizinhanças principais e imediatas desse ponto estão mostradas na Fig. 1, que mostra o relacionamento dos símbolos do *chain code* com a vizinhança-4 e vizinhança-8.

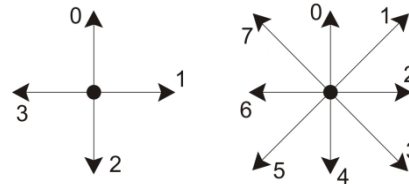


Figura 1. À esquerda estão os símbolos de 0 a 3 do *chain code* para vizinhança-4. À direita, os símbolos de 0 a 7 do *chain code* para vizinhança-8 (adaptado de [21]).

Um arco digital é reto se for o resultado da digitalização de um segmento em linha reta euclidiano. No modelo de Freeman, as cadeias ideais que representam linhas retas obedecem a três propriedades, em uma codificação utilizando números de 0 até 7, em vizinhança-8:

--Prop1: No máximo dois tipos de símbolos, representando direções distintas no código do *chain*, podem estar presentes, e estes são números consecutivos correspondentes do *chain*, módulo oito;

--Prop2: Um dos dois símbolos sempre ocorre isoladamente, solitário;

--Prop3: As ocorrências sucessivas do símbolo isolado são tão uniformemente espaçadas quanto possível entre códigos do outro valor, que ocorre em grupos ou corridas (runs).

O significado de Prop1 a Prop3 é representar a linha por uma sequência de vetores com inclinações múltiplas de 45° e cujos comprimentos são unitários (se horizontal ou vertical) ou $\sqrt{2}$ (se diagonal).

A noção de linha reta está associada à propriedade da

corda, pois [17] demonstrou que a condição necessária e suficiente para que um arco digital qualquer seja um SLRD ideal é atender à propriedade da corda.

Definição 2: A Propriedade da Corda: Diz-se que um arco digital C , representando “objetos sólidos” delgados em uma imagem digitalizada, apresenta a propriedade da corda se, para cada dois pontos digitais c e d pertencentes a C , e para cada ponto $p = (x, y)$ em $\overline{cd}$, existe um ponto $e = (h, k)$ pertencente a C tal que $\max\{|x - h|, |y - h|\} < 1$, onde $\overline{cd}$ é o segmento de reta entre c e d [17].

A Definição 2 implicou na demonstração da existência de uma estrutura hierárquica composta de números consecutivos correspondentes às corridas dos símbolos especificados por Prop1 e Prop2. Essa estrutura de números consecutivos é expressa por uma propriedade adicional Prop4: Quanto à direção referente ao símbolo que ocorre em grupos (não isolado), as corridas correspondentes podem ocorrer com apenas dois valores, os quais diferem de uma unidade (por exemplo, P e $P+1$).

Pela propriedade da corda, define-se um SLRD ideal como a seguir.

Definição 3: Linha Reta Digitalizada Ideal (SLRD ideal): Uma linha reta digitalizada ideal é aquela que atende à propriedade da corda, na vizinhança respectiva.

Também os conceitos de unidade de segmento e ângulo de orientação são importantes para o entendimento das propriedades locais e globais dos SLRD.

Definição 4: Unidade de Segmento Digital em Linha Reta (abreviado USLR) é o menor segmento possível em que um SLRD pode ser subdividido a fim de manter o correspondente ângulo de orientação principal [20].

Definição 5: Ângulo de Orientação Principal: Entende-se como ângulo de orientação principal do SLRD, denominado θ_s , como aquele ângulo que se destaca relativamente ao eixo x , por algum critério, dentre a distribuição de ângulos das USLR individuais do segmento.

O processo de digitalização de uma linha reta euclidiana específica resulta que o correspondente SLRD não atenda à propriedade da corda, resultando em muitos segmentos curtos na vizinhança de um pixel, devido às influências espúrias. Portanto, é necessário algum tipo de medida, denominada métrica, apropriada para avaliar se dois SLRD pertencem a uma única estrutura linear [18] [22]. Por essa métrica, pode-se definir uma função de vizinhança escolhida de modo a expressar como agrupar SLRD aproximados.

Contudo, “aproximado” é entendido no tocante a linhas retas visualmente corretas dentro de uma tolerância, não necessariamente no tocante à propriedade da corda. Portanto, de agora em diante, a menos que de outra maneira seja especificado, a nomenclatura deste texto não faz nenhuma distinção entre um SLRD ideal e as linhas “quase” retas nas proximidades da primeira, considerando que tanto SLRD ideais quanto SLRD aproximados são reconhecido pelo dispositivo adaptativo deste trabalho.

B. Codificação

Nesta pesquisa, os SLRD não são codificados por meio de

cadeias de dígitos, bastando considerar $a = 0$ e $b = 1$ no segmento exemplificado pela Fig. 2 a fim de atender Prop1. Uma cadeia é uma sequência de zero ou mais símbolos pertencentes ao alfabeto Σ . Esses símbolos são também denominados como primitivas, *tokens*, elementos do *chain code* ou simplesmente estímulos. O conjunto de todas as cadeias possíveis com Σ é denotado por Σ^* . O comprimento de uma cadeia qualquer S é denotado por $|S|$. A cadeia vazia, de comprimento zero, é representada por ϵ . O i -ésimo símbolo de uma cadeia $S = s_1 \dots s_n$ é representado por s_i .

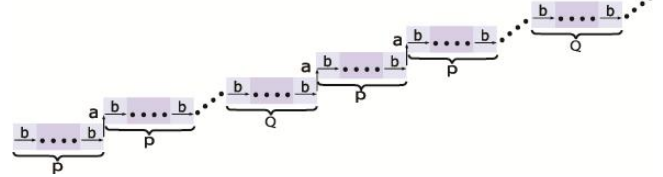


Figura 2. Exemplo de SLRD genérico na faixa $0 < \theta_u < \pi/4$ (θ_u é o ângulo de inclinação relativo ao eixo x de USLR), com corridas de P e Q símbolos b , tão uniformemente espaçados quanto possível entre *tokens* a , que ocorrem isolados (adaptado de [20]).

Caso nada em contrário seja especificado, sem qualquer redução em generalidade, a vizinhança-4 é utilizada neste trabalho como padrão, com os símbolos da propriedade Prop1 consecutivos, módulo quatro. Mais precisamente, os símbolos que compõem as cadeias pertencem a $\Sigma = \{a, b, c, d\}$ tal que, para atender Prop1, basta considerar módulo 4, juntamente com $a=0, b=1, c=2, d=3$, relativamente à vizinhança-4 da Fig. 1.

Uma cadeia qualquer $S = s_1 \dots s_n$ pode também ser representada no formato da Expressão 1.

$$S: s_i; i = 1, 2, \dots, n. \quad (1)$$

Caso todos os elementos de S da Expressão 1 sejam idênticos, $S = s_1 = s_2 = \dots = s_{n-1} = s_n = s$, uma representação compacta é

$$S = s^n. \quad (2)$$

Por exemplo, na Fig. 2, as cadeias das USLR são do tipo ab^n com n inteiro, $n=0, 1, 2, 3, \dots$; resultando a expressão para o ângulo $\theta_u = \arctang(1/n)$.

C. O Autômato Finito Adaptativo (AFA)

Definição 1: O Autômato Finito Adaptativo (AFA) é um dispositivo auto-modificável, com poder computacional equivalente à máquina de Turing, da forma $AFA = (ND0, AM)$ onde ND0 é a camada subjacente, e AM é a camada adaptativa associada, formalizada nos mesmos moldes que ND0. O AFA é caracterizado por ter o autômato finito (AF) como formalismo da camada subjacente. Bibliografia e evolução do AFA e adaptatividade estão em [32].

A camada adaptativa da Definição 1 integra o conjunto de ações adaptativas, responsáveis por alterar a camada subjacente dinamicamente em resposta aos estímulos. As ações adaptativas podem ser interpretadas como chamadas de função, a função adaptativa (FAD), em que esta pode ser paramétrica.

A Fig. 3 mostra uma representação gráfica estática genérica de transição do AFA do tipo: $(x, i) : R \rightarrow y : S$ onde x é o estado atual do autômato antes da transição entre estados; y é o estado de destino do autômato após a transição; i é o estímulo

de entrada; R e S são as FAD dos tipos anterior e posterior respectivamente por serem ativadas “antes” e “depois” da transição do estado x para o estado y .

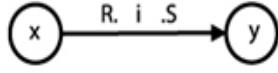


Figura 3. Uma transição adaptativa genérica, onde R e S são FAD opcionais do tipo anterior e posterior, respectivamente.

Por sua vez, as funções adaptativas são formadas por conjuntos de ações adaptativas elementares ou primitivas a serem aplicadas para definir o processo adaptativo do autômato. Pela Tab. I, tais ações adaptativas elementares podem ser de três modalidades, de acordo com o prefixo de uma transição especificada entre colchetes, denominada gabarito. Por esse gabarito, as transições em uso devem ser testadas por um processo de busca da seguinte maneira:

- 1) Ações de consulta, denotadas pelo prefixo “?”: efetuam uma busca por regras que casem com o gabarito especificado, sem alterar o conjunto de regras;
- 2) Ações de remoção, denotadas pelo prefixo “-”: removem do conjunto as regras que casam com o gabarito especificado;
- 3) Ações de inserção, denotadas pelo prefixo “+”: inserem o gabarito especificado no conjunto de regras.

TABELA I
FORMATO DAS AÇÕES ADAPTATIVAS ELEMENTARES.

MODALIDADE	PREFIXO	SIMBOLOGIA
CONSULTA	?	$?[(x, i) : R \rightarrow y : S]$
REMOÇÃO	-	$-[(x, i) : R \rightarrow y : S]$
INSERÇÃO	+	$+[(x, i) : R \rightarrow y : S]$

Observe-se que, havendo mais de uma ação adaptativa elementar a ser executada, independentemente da ordem em que foram declaradas, têm precedência as consultas. Em seguida são efetuadas as remoções, e posteriormente as inserções. Possíveis transições em vazio são sempre executadas por último, após as inserções. Ações elementares que referenciam elementos indefinidos não são consideradas.

C1. Formato das Funções Adaptativas (FAD)

Opcionalmente, as FAD podem ser paramétricas. Apesar de opcionais, se parâmetros forem especificados, terão de ser fornecidos para ativar a correspondente FAD. Em outras palavras, os parâmetros formais de FAD são variáveis especiais, utilizados para que valores reais (argumentos) sejam atribuídos, sempre que determinada FAD seja ativada. Assim, aos parâmetros da FAD são atribuídos os valores recebidos como argumentos antes de ativar a FAD, permanecendo inalterados durante toda a sua execução.

No caso geral, além dos parâmetros, uma FAD pode ser especificada por variáveis e geradores, também opcionais, com os significados seguintes:

Variáveis: identificadores que recebem um valor de acordo com o resultado de ações adaptativas elementares de consulta ou exclusão. As variáveis são utilizadas para representar entidades existentes em transições do AFA tais como estados

ou estímulos.

Geradores: identificadores que recebem um valor único (isto é, nunca antes utilizado) no início da ação adaptativa e permanecem com este valor até o final da ação. São espécies de variável que contém a identificação do próximo estado a ser incluído na topologia do AFA.

Graficamente, uma determinada FAD R é representada por R. (R seguido de ponto), caso seja do tipo anterior. Similarmente, uma FAD S é representada por .S (ponto seguido de S), caso seja do tipo posterior.

D. O Segmento Digitalizado Adaptativo

Em um primeiro aspecto, SLRDA são implementados nesta pesquisa por meio de AFA a fim representar as diferentes instâncias do modelo ideal de SLRD afetada pelos desvios em ângulo. Isso requer que o SLRDA atue numa determinada faixa de ângulos, definindo uma vizinhança, para comprimentos de arcos digitalizados teoricamente quaisquer. Tal vizinhança associa um conjunto de arcos a um SLRDA correspondente, que reconhece esse conjunto.

Em um segundo aspecto, SLRDA são implementados também relacionados a uma vizinhança adaptativa, tal que as vizinhanças dos SLRDA são alteradas adaptativamente em função dos respectivos comprimentos dos arcos.

A Fig. 4 mostra um SLRDA, tal que com o primeiro símbolo b , a FAD B é ativada, alterando-se a configuração do AFA para a Fig. 5, passando o AFA a consumir as USLR subsequentes, sendo que cada USLR da cadeia de entrada ativa a FAD RB. Ou seja, com o consumo da $USLS_1$, recebendo-se o primeiro símbolo b , a configuração do AFA altera-se para a Fig. 5. Na Fig. 5, da mesma maneira que na Fig. 4, com o acionamento da FAD RA, uma transição em vazio é removida. A FAD RB insere novamente as transições em vazio, preparando o AFA para o recebimento de outra USLR.

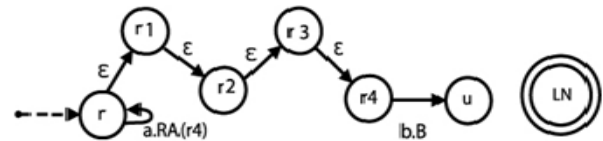


Figura 4. Configuração inicial de AFA que implementa SLRDA a fim representar as diferentes instâncias do modelo ideal de SLRD afetada pelos desvios em ângulo. A cada ativação de RA, uma das transições em vazio é removida, reconhecendo a primeira unidade de segmento: $USLR_1$.

Ainda na Fig. 5, especificando-se a quantidade de estado do loop (exemplificado pelos estados $u, u_1, u_2, u_3, u_4, u_5$) e a quantidade de transições em vazio esse autômato foi implementado para que reconheça cadeias de entrada com USLR do tipo $USLS_i = \{a^nb : 3 \leq n \leq 5\}$, com i inteiro, $i = 0, 1, 2, 3, \dots$

Observe-se que transições em vazio causam não determinismo, podendo-se implementar um autômato determinístico equivalente sem transições em vazio, alterando-se o AFA das Fig. 4 e 5 utilizando-se transições com marcadores tal como Δ sendo que $\Delta \notin \Sigma$.

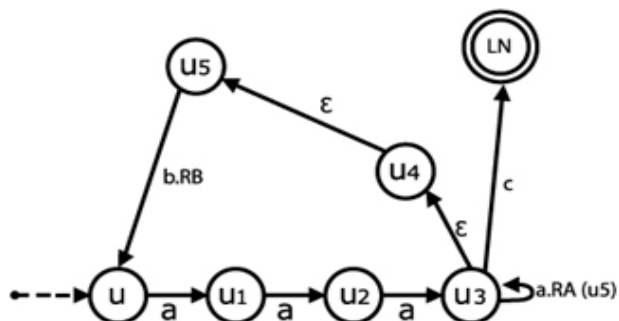


Figura 5. Configuração de AFA após ativar a FAD B da Fig. 4. A faixa de ângulos reconhecida por este SLRDA é ajustada pela quantidade de transições em vazio e transições que consomem o símbolo a que compõem o *loop* compostos, como exemplo, pelos estados u , u_1 , u_2 , u_3 , u_4 , u_5 .

III. ENUNCIADO DO PROBLEMA

O presente trabalho investiga um novo modelo de representação computacional do ambiente por um robô, tarefa considerada fundamental para a robótica móvel conforme trabalhos relevantes da área citados no tópico I. Dentre os trabalhos comentados no tópico I, [2] relata vários algoritmos de navegação em um ambiente envolvendo, normalmente, leitura do mapa representativo do ambiente ou espaço de trabalho e, subsequentemente, tentar criar caminhos livres para que o robô percorra a área de trabalho sem colidir com os objetos e obstáculos. Existem métodos tradicionais de representar o ambiente para navegação. Por tais métodos, os ambientes tendem a não serem representados de forma ideal, pois normalmente geram um mapa abstrato representativo correspondendo a uma grade fixada *a priori* por coordenadas cartesianas. O assunto se relaciona com os temas clássicos da geometria computacional e do planejamento de movimento e navegação [6], tendo em vista que um percurso a ser percorrido por um agente robótico é composto por pontos interligando um local de partida com uma localização final, compondo uma rota de referência. Considerando as diversas aplicações existentes na atualidade, a linha reta, interligando o ponto de partida e o ponto final, pode ser considerada como a melhor rota, a título de referência dos algoritmos de navegação a serem implementados por este trabalho.

No contexto do planejamento de rotas em navegação, o agente robótico deve detectar os elementos em seu ambiente e o significado de cada situação tática para definir rotas alternativas. Ou seja, o agente robótico utiliza restrições de espaço, localização do alvo ou ponto no espaço a alcançar pela navegação e obstáculos existentes a serem evitados durante o movimento para definir alterações da rota de referência. Na reavaliação dinâmica do ambiente, o agente robótico tem que se adaptar a uma nova rota reiteradamente, sempre que ocorra alteração do ambiente ou mudança de situação tática.

A. Desafios científicos e tecnológicos

Os desafios científicos e tecnológicos desta pesquisa podem ser relacionados às áreas envolvidas com o trabalho proposto, principalmente reconhecimento de padrões, geometria computacional ou geometria digital e compiladores, comentados a seguir:

A1. Desafios científicos

Os desafios científicos são devido ao ineditismo da proposta que é o uso do formalismo de SLRDA para representar ambientes e o movimento de robôs em navegação, como alternativa ao modelo cartesiano. Destaque-se que, apesar de existir na literatura uma infinidade de trabalhos sobre arcos e retas digitalizados (SLRD), que tiveram início nas proximidades da década de 1970, constata-se que os resultados obtidos com os métodos sintáticos em áreas diversas tendem a apresentar restrições quando relacionados a cenários sujeitos a influências espúrias (erros e ruído), tais como em navegação robótica, restringindo os trabalhos, por exemplo, às linguagens regulares [19] [23].

Em particular, ao longo dos últimos anos, os estudos em planejamento de trajetórias ou caminhos têm aumentado sensivelmente, pois robôs móveis são hoje utilizados em variadas aplicações envolvendo operar em presença de incerteza em vários domínios tais como atuação nas áreas industriais e residenciais [5], as operações de resgate em situações de catástrofe, exploração marítima e planetária [24], defesa e a execução de tarefas agrícolas [25][1].

Essas modernas aplicações implicam em utilizar os avanços tecnológicos em sistemas digitais. Considerando que as modelagens de sistemas por equações de diferenças e diferenciais há muito tempo têm fornecido um quadro rico para a análise, projeto e controle dos processos, a característica orientada a eventos da tecnologia digital conduz a um comportamento dinâmico em que a abordagem centrada na modelagem de sistemas por equações de diferenças e diferenciais é simplesmente inadequado [11], se não considerar os erros envolvidos entre as modelos contínuos e discretos e as perturbações espúrias ou ruído inerentes à aplicação.

A2. Desafios tecnológicos

Há 3 vertentes principais de desafios tecnológicos para a navegação robótica no contexto apresentado. A primeira é de natureza de obtenção de informações do mundo real: definir a trajetória, identificar os pontos críticos na trajetória e ângulo de curvaturas locais, para finalmente efetuar ainda uma análise global da trajetória para detectar o comprimento de subtrajetórias entre pontos críticos, descartando ou não ângulos de curvatura abaixo de determinado valor. A outra vertente é a dos desafios envolvidos no modelo para representação do ambiente: alterar o modelo cartesiano com granularidade fixa especificada *a priori*, para uma representação dinâmica do ambiente com granularidade que se altera adaptativamente de acordo com os estímulos.

A terceira é a dos desafios envolvidos na atualização dinâmica das informações e das representações, face aos estímulos e obstáculos, levando em conta o movimento do robô pelo ambiente, em que o processo de digitalização é afetado por erros na fonte de dados, de erros no processamento anterior à transformação para o formato de cadeia, ou até mesmo erros na transformação final para cadeia, e por ruído em todo o processamento. Outras fontes de ruído e erros são causadas por efeitos como qualidade da imagem não ideal, alterações do foco da câmera afetando as dimensões e ângulos

de segmentos, pré-processamento imperfeito, distorções afetando as escalas, aproximações em modelos e equações. Como resultado, a navegação por SLRDA impõe formalismos de gramáticas irrestritas [15], tipo 0, integradas a cenários em robótica sujeitos a distorções e ruído.

Esclarecendo melhor sobre o poder computacional requerido por este trabalho, os SLRD implicam em linguagens tipo 1, entretanto, devido aos comprimentos de segmentos representado pelos SLRDA, que podem ser em diferentes escalas, tendendo teoricamente até infinito, esta pesquisa impõe formalismos com poder computacional equivalentes à da máquina de Turing (detalhes adicionais sobre esse aspecto estão em [14]).

B. Meios e métodos para superar os desafios.

Os meios e métodos para superar os desafios são os seguintes:

B.1 Superação dos desafios científicos

A superação dos desafios científicos se efetua aplicando o formalismo adaptativo apresentado por [14] para o registro e tratamento das informações adquiridas pelo conjunto de sensores de um robô móvel. A arquitetura apresentada em [26] é aperfeiçoada por esta pesquisa para incluir os registros de dados de erros e ruído em rotas de qualquer ângulo relativamente a uma direção de referência ou azimute, conforme esta proposta.

O formalismo de [14] é aplicado na superação dos desafios científicos principalmente nos seguintes pontos: i) Utilizar o formalismo de SLRDA para representação não cartesiana do ambiente e do movimento do robô, que considere a curvatura e comprimento dos arcos de navegação; ii) Aplicar a adaptatividade para representar os erros, as tolerâncias e os parâmetros envolvidos; iii) Utilizar uma vizinhança adaptativa a fim de incorporar dinamicamente ao modelo as tolerâncias e erros nos parâmetros, tais como ângulo e comprimento, otimizando a capacidade de representação computacional da informação; iv) Utilizar a facilidade de escalas adaptáveis para se obter trajetórias de navegação que podem ser em diferentes escalas, com comprimentos relativos tendendo teoricamente até infinito.

A escala adaptável possibilita alterar a representação geométrica discreta do ambiente e do movimento do robô, alterando os erros de quantização adaptativamente em função dos comprimentos e curvaturas dos arcos de navegação. Com esse procedimento, detalhes detectados para um determinado espaçamento da grade representativa do ambiente, não serão detectados em outros, e vice-versa. De acordo com a precisão requerida, a grade digital mencionada deve ser suficientemente fina para preservar o raio de curvatura mínimo dos arcos das trajetórias.

B.2 Superação dos desafios tecnológicos

Com relação aos agentes robóticos propriamente ditos, o presente trabalho emprega AFA para definição da trajetória de robôs, nos moldes de [26], o qual relaciona várias referências sobre pesquisas que empregam máquinas de estados e autômatos, isoladamente ou conjuntamente a outras funcionalidades da robótica móvel. Em outras palavras, determinados AFA neste trabalho estarão associados a agentes

robóticos em navegação, tendo em vista que [14] demonstrou a viabilidade em aplicar o formalismo dos AFA em geometria digital para modelagem de SLRD.

De conformidade com esse processo de associar agentes robóticos a AFA, a superação dos demais desafios tecnológicos se realiza por simulações, correspondentes a provas de conceito objetivando a validação, teste e análise dos resultados obtidos com a proposta. A escolha dos ambientes simulados e rotas de navegação de referência utilizados nas provas de conceito se baseiam na representatividade em função de características específicas visando abranger grande parte das situações possíveis.

IV. SÍNTESE DO MÉTODO PROPOSTO

Este tópico apresenta detalhes técnicos sobre o comentado no item anterior.

Quanto à modelagem de software, este é representado como um sistema de máquinas de estados para cobrir o espaço ambiental do robô, esquema adaptado de [36]. Essa modelagem é conveniente também devido à necessidade de disponibilizar um banco de autômatos (composto de AFA e AF) para cobrir a faixa de ângulos, variável de 0 a 360 graus, para as trajetórias possíveis de robôs. A quantidade de SLRDA implementados em cada setor depende de escolha considerando critérios tais como precisão requerida nas trajetórias.

Observe-se que os autômatos permanecem invariáveis com alterações adaptativas da grade, pois esta, em última análise, envolve alterar adaptativamente o comprimento das primitivas da Fig. 1, relativamente às dimensões efetivas do ambiente em que está inserido o robô.

A Tabela II mostra 8 setores que cobrem 2π radianos (rad), em que cada setor corresponde a SLRDA, cuja expressão de USLR está também indicada na tabela, com n inteiro $n=0, 1, 2, \dots$

TABELA II
FAIXAS DE ÂNGULOS θ_u RELATIVO AO EIXO X DE SLRDA

SETOR	EXPRESSÃO DE USLR	FAIXA DE ÂNGULO (RAD)
1	ab^n	$0 < \theta_u < \pi/4$
2	ba^n	$\pi/4 < \theta_u < \pi/2$
3	da^n	$\pi/2 < \theta_u < 3\pi/4$
4	ad^n	$3\pi/4 < \theta_u < \pi$
5	cd^n	$\pi < \theta_u < 5\pi/4$
6	dc^n	$5\pi/4 < \theta_u < 3\pi/2$
7	bc^n	$3\pi/2 < \theta_u < 7\pi/4$
8	cb^n	$7\pi/4 < \theta_u < 2\pi$

Para os ângulos de USLR ($0 + \frac{k\pi}{2}$) radianos, com k inteiro $k=0, 1, 2, \dots$; as cadeias representando segmentos de retas são do tipo $aaaaaa....$; $bbbbbb....$; $cccccc....$; $dddddd....$; ou seja, são reconhecíveis por 4 AF.

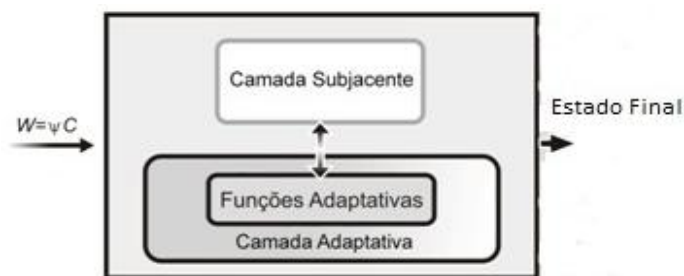
Para os ângulos ($\frac{\pi}{4} + \frac{k\pi}{2}$) radianos, com k inteiro $k=0, 1, 2, \dots$; as cadeias representando segmentos de retas são do tipo $ababababab....$; $bcbcbcbcbcb....$; $dcdcdcdcdcd....$;

Complementando o comentado em II.B, as faixas de ângulo da Tabela 2, são obtidas pelas cadeias das USLR. Por exemplo, no setor 1, as USLR são do tipo ab^n com n inteiro, $n=0, 1, 2, 3, \dots$; resultando que o ângulo $\Theta_u = \arctang(1/n)$. Para o setor 2, USLR são do tipo ba^n implicando $\Theta_u = \arctang(n)$. E assim por diante para todos os setores.

V. IMPLEMENTAÇÃO

- O Sistema Gerente é uma camada adaptativa hierarquicamente superior, que tem a função de monitorar e coordenar SLRDA, sendo que a quantidade de SLRDA altera-se conforme os estímulos pelo gerenciamento dessa camada. O Sistema Gerente recebe as informações dos SLRDA pelos correspondentes estados finais, bem como pela informação se a cadeia de entrada foi aceita ou rejeitada;
- O Sistema Escravo é a camada adaptativa comandada pelo Sistema Gerente ou camada adaptativa superior, composta por um ou mais SLRDA.

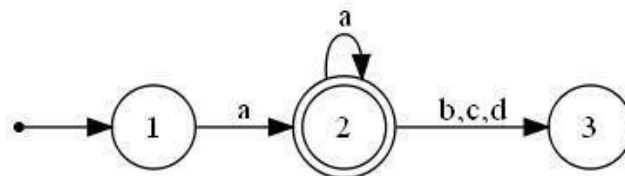
Na Fig. 6, Ψ é uma cadeia recebida pelo AFA para que este ajuste uma faixa de tolerância relacionada à cadeia C . Ψ corresponde a um fator para compensar imprecisões no processo de digitalização. O assunto de imprecisões e erros em digitalização é aprofundado em [14], o qual relaciona vários trabalhos sobre essa questão.



A. Implementação de autômatos

Um dos padrões de projeto é o SLRDA do tópico II.D implementado para reconhecimento de SLRD submetidos a erros em inclinação (ângulo relativo ao eixo x), exemplificando para cadeias de entrada com USLR do tipo $USLS_i = \{a^n b : 3 \leq n \leq 5\}$, com i inteiro, $i = 0, 1, 2, 3, \dots$

Outros exemplos de padrões de projeto são os seguintes, utilizados para implementação dos AF descritos na tópico IV-Síntese. i) O AF que reconhece cadeias do tipo *aaaaaaa...* da Fig. 7; ii) O AF que reconhece cadeias do tipo *abababababab.....* da Fig. 8.



```

graph LR
    start(( )) --> 1((1))
    1 -- a --> 2((2))
    1 -- b --> 3((3))
    2 -- b --> 4(((4)))
    3 -- a --> 5(((5)))
    4 -- a --> 6(((6)))
    6 -- b --> 4
    5 -- b --> 7(((7)))
    7 -- a --> 5
  
```

Figura 8 AF que reconhece cadeias do tipo *abababababab.....*

B. O banco de autômatos

Cada setor da Tab. II contribui com quatro AFA, denominados A, B, C e D. Por exemplo, para o primeiro quadrante composto pelo setor 1 e setor 2 da Tab. II, os seguintes SLRDA foram selecionados:

Setor 1 (ab^n): $2 \leq n \leq 4$ (Autômato A); $5 \leq n \leq 7$ (Autômato B); $8 \leq n \leq 12$ (Autômato C); $13 \leq n \leq 20$ (Autômato D).

Setor 2 (ba^n): $2 \leq n \leq 4$ (Autômato A); $5 \leq n \leq 7$ (Autômato B); $8 \leq n \leq 12$ (Autômato C); $13 \leq n \leq 20$ (Autômato D).

Para os demais setores a topologia dos AFA é a mesma dos setores 1 e 2. Resulta em 32 AFA e 8 AF, totalizando um banco de 40 autômatos.

VI. JUSTIFICATIVA PARA OS EXPERIMENTOS

Este tópico apresenta justificativas para o método de comparação nos experimentos, comparando-se com trabalhos selecionados, posicionando esta pesquisa com relação ao estado da arte, descrita a seguir.

A. Posicionamento com relação ao estado da arte.

Com referência ao comentado no item I, destaque-se que as trajetórias de robôs requerem ângulos variáveis de trajetórias (denominada por [34] como “navegação em qualquer ângulo”). Assim, [34] afirma que as trajetórias relativas à grade cartesiana tendem a não serem os caminhos mais curtos (isto é, os caminhos mais curtos no terreno), pois se restringem artificialmente a ângulos múltiplos de 45 graus (para vizinhança-8) e múltiplos de 90 graus (para vizinhança-4). Em outro trabalho com o mesmo propósito de ângulos variáveis em trajetórias, [36] formaliza um espaço de busca de rotas alternativas por meio de algoritmo de busca em grafos.

Quanto às técnicas adaptativas utilizando autômatos, certamente o estado da arte é representado pelo trabalho [26] que se restringe a trajetória em apenas quatro direções (norte, sul, leste e oeste), suficientes para o propósito de mapeamento de ambientes. A motivação de [26] em utilizar a adaptatividade foi devido à elevada quantidade de memória para registro de informações para mapeamento de ambientes, implicando em alto poder computacional de seus sistemas de planejamento de trajetória.

Como breve resumo desse trabalho, [26] utiliza dois AFA, um deles denominado Autômato de Exploração, em que a informação de saída indicada pela execução deste autômato determina qual a direção do próximo movimento a ser executado pelo sistema motor do robô, nas quatro direções. Em sequência, tal informação é transmitida para o outro AFA, denominado Autômato de Mapeamento, responsável pelo registro dos dados sobre a topologia do ambiente que está sendo mapeado.

Pelo registro topológico do ambiente, constrói-se, por estados de AFA: i) Uma trajetória de retorno de um local já conhecido que envolva um menor número de deslocamentos do robô durante o processo de exploração; ii) O mapa do ambiente construído durante o processo de navegação do robô, registrado na memória do agente robótico.

A metodologia desta pesquisa envolve também a formalização de um espaço de busca de rotas com certa analogia com [36], porém tal espaço é representado nesta pesquisa por uma vizinhança adaptativa, por estados do AFA. Este estudo propõe aplicar o formalismo dos SLRDA para possibilitar navegação em qualquer ângulo, atendendo ao requerido por [34] e [36], integrando o ponto positivo obtido por [26] por meio da adaptatividade viabilizando a utilização progressiva da memória do sistema, consumida de acordo com a área já mapeada.

Complementarmente, o formalismo dos SLRDA apresenta vantagens adicionais em utilização de memória, considerando que basta o registro de coordenadas de apenas dois pontos, os pontos extremos dos segmentos digitalizados representando trajetórias, e não mais o registro de todos os pontos das trajetórias para construção do mapa, conforme [26].

Pelo fato de que as trajetórias variam em ângulo com qualquer comprimento, em que este comprimento é variável teoricamente até infinito, inviabiliza-se a utilização de autômatos finitos com os propósitos deste estudo. Resulta que esta proposta se posiciona na pesquisa de dispositivos e algoritmos guiados por regras (com adaptatividade hierárquica multi-nível) cujo conjunto de regras é variável apresentando funções adaptativas modificáveis [32].

B. O Método Clássico

Os experimentos utilizam trajetórias variáveis em ângulo e comprimento, comparando-se com o método clássico em que se inclui o trabalho de [26]. Isso porque a denominação “método clássico” significa neste trabalho a não utilização da adaptatividade, ou adaptatividade parcial por existência de restrições.

Desse modo, o método clássico é constituído normalmente por uma ou mais das seguintes características dos dispositivos ou algoritmos: não utilização da adaptatividade, movimento do robô em apenas umas poucas direções (comumente quatro direções ortogonais ou oito direções pela inclusão das direções diagonais, tais como as direções definidas pela vizinhança-4 ou vizinhança-8, respectivamente), registro de todos os pontos das trajetórias para construção do mapa, comprimentos das trajetórias limitados, indefinidos ou não formalizados, recuperação de erros em trajetórias não formalizada.

Assim, levando-se em conta que [26] representa o estado da arte quanto às técnicas adaptativas utilizando autômatos, mais especificamente o AFA, os experimentos são comparados com esse trabalho. Dessa forma, o método de comparação usado nos experimentos mostra a trajetória por [26], composto de pequenos movimentos nas quatro direções ortogonais (norte, sul, leste e oeste) e a trajetória pelo método proposto resultante de cadeia de entrada para AFA atuando como autômato de movimento.

Por meio dessas comparações, chega-se a conclusões quanto à robustez da representação de ambientes de navegação de robôs móveis por segmentos de linhas retas digitalizadas (SLRD) pelo modelo de SLRDA, baseado em AFA.

VII. EXPERIMENTOS

Os experimentos de [14] para ensaios do SLRDA das Fig. 4 e 5 utilizaram as cadeias de entrada da Fig. 9, especificadas na Tab. III, mostrando o reconhecimento adequado

independentemente de desvios em ângulo (em uma determinada faixa de tolerância) e variações em comprimento.

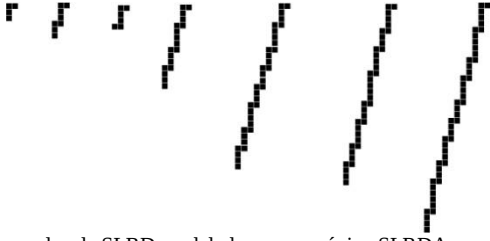


Figura 9. Exemplos de SLRD modelados por um único SLRDA.

TABELA III
CADEIAS DA FIGURA 9 (DA ESQUERDA PARA A DIREITA)
MODELADAS POR UM ÚNICO SLRDA.

CADEIAS DA FIG.	CODIFICAÇÃO
7	
1	a^2b
2	a^2ba^3b
3	ba^3b
4	$a^3ba^3ba^4ba^3b$
5	$a^3ba^3ba^4ba^3ba^3ba^4ba^3b$
6	$a^3ba^3ba^4ba^3ba^5ba^4ba^3ba^4b$
7	$a^3ba^5ba^3ba^4ba^3ba^3ba^4ba^3ba^5ba^4b$

Os experimentos deste trabalho apresentam analogia aos de [14], integrando-se o movimento do robô. O deslocamento do robô é previsto inicialmente por este trabalho como da ordem da dimensão do robô, ou seja, esse fato envolve uma questão de escala compatível entre cada passo para movimento do robô e as suas respectivas dimensões.

A. Comparação de trajetórias com o método clássico

A Fig. 10 mostra em tracejado a trajetória percorrida pelo robô pelo método clássico, o qual é apresentado numa escala não compatível com o passo de seu deslocamento, definido pela grade.



Figura 10. O robô é representado pela bola preta, fora de escala compatível com o passo de sua trajetória, indicada pelo tracejado.

A Fig. 11 mostra em pontilhado a trajetória pelo método clássico do mesmo robô, em duas posições diferentes, indicado em escala compatível com o passo da trajetória. A trajetória por esta pesquisa é representada pela linha reta.

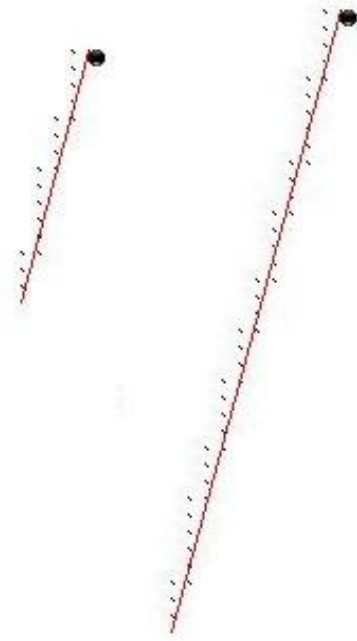


Figura 11. A figura mostra o mesmo robô em duas posições diferentes da mesma trajetória. O robô se desloca em trajetória indicada pelos pontos pelo método clássico [26]; enquanto que, pelo método proposto por esta pesquisa, o robô se desloca em trajetória representada pela linha reta. A figura à esquerda mostra o robô em uma posição intermediária e, à direita, a posição final para a cadeia de entrada $a^3ba^5ba^3ba^4ba^3ba^3ba^4ba^3ba^5ba^4b$, reconhecida pelo AFA das Fig. 4 e Fig. 5.

Uma maneira didática para descrever o método proposto por esta pesquisa é considerar que o robô se move a cada USLR da cadeia de entrada (correspondente a um SLRD), ao invés de se mover em cada símbolo pelo método clássico. Implica que o robô percorra uma trajetória composta de pontos correspondentes a cada USLR. Resulta que, dada SLRD de entrada com n USLR:

$\{USLR_1, USLR_2, USLR_3, USLR_4, USLR_5, \dots, USLR_n\}$, o robô terá n deslocamentos curtos até atingir o ponto final.

Outro modo, computacionalmente mais efetivo que o anterior, é o robô se direcionar diretamente ao ponto final, pois, neste caso envolve a troca de informações com o robô apenas de coordenadas indicativas da localização no mapa do ambiente de apenas um ponto. Isso é esquematizado na Fig. 11 pelas linhas retas, representado a trajetória pelo método proposto.

B. Grade Adaptativa

A Fig. 12 mostra pelas linhas retas as trajetórias percorridas pelo robô, em duas grades diferentes. Nesse caso de alteração adaptativa da grade, o banco de autômatos permanece inalterado.

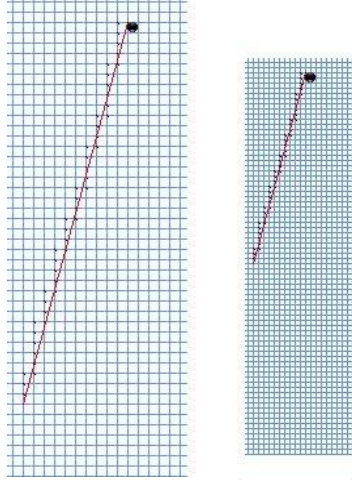


Figura 12. As linhas retas são exemplo de trajetórias pelo método proposto por esta pesquisa, em duas grades diferentes. A cadeia de entrada SLRD = $a^3ba^5ba^3ba^4ba^3ba^4ba^3ba^4b$ é reconhecida pelo AFA das Fig. 4 e Fig.5.

C. Segmentação de trajetórias por segmentos adaptativos em escalas adaptáveis

A Fig. 13 mostra um robô em uma trajetória segmentada em três SLRD (cadeia 1, 2 e 3).

A cadeia 1 é $SLRD_1 = a^2ba^4ba^3ba^4ba3ba^3ba^4ba^4b$ reconhecida pelo Autômato B-setor 2.

A cadeia 2 é $SLRD_2 = (cb^5)(cb^6)(cb^7)$ reconhecida pelo Autômato B – setor 8. A cadeia 3 é $SLRD_3 = (c^8b)(c^9b)(c^{12}b)^4$ reconhecida pelo Autômato C-setor 7.

Na Fig. 13, no centro, as cadeias estão em uma mesma escala. Na Fig. 13, à esquerda e à direita, as escalas da cadeia 3 são 0,5 e 1,5 respectivamente.

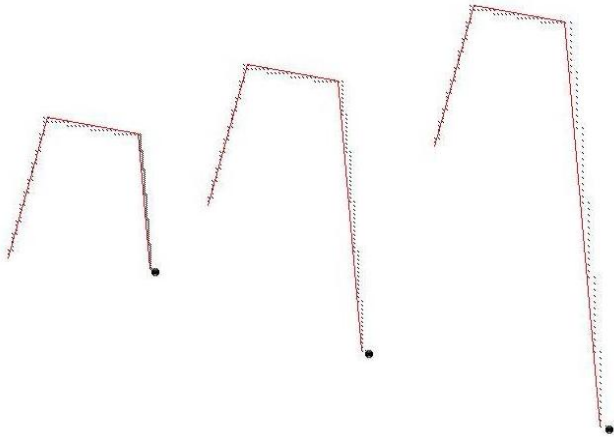


Figura 13 A figura mostra uma mesma trajetória, segmentada em 3 SLRD (cadeia 1, cadeia 2 e cadeia 3), com a cadeia 3 em grades diferentes. As trajetórias pelo método clássico são indicadas pelos pontilhados, enquanto as linhas retas mostram as trajetórias pelo método desta pesquisa.

VIII. TRABALHOS FUTUROS

Trabalhos futuros envolvem estudos de algoritmos para navegação robótica em cenários com obstáculos, investigando a necessidade de SLRDA de exploração e mapeamento. O SLRDA de Mapeamento armazena os dados indicativos da trajetória que está sendo percorrida pelo conjunto de unidades, principalmente comprimentos dos arcos; O SLRDA de Exploração tem a função de atuar nas mudanças de direção do próximo deslocamento efetuado pelo robô, durante a exploração da trajetória prevista, decidindo favoravelmente ou

não pela mudança. Por exemplo, no caso de uma região de alta curvatura e comprimento muito pequeno relativamente ao arco global, o SLRDA de Exploração evitaria a mudança de direção. Esta decisão é feita com base em regiões de suporte variáveis adaptativamente, dos dados coletados sobre a existência ou não de mudanças de curvaturas ou direções, nas informações previamente registradas pelo SLRDA de Mapeamento e no algoritmo de exploração executado por este autômato.

Quanto aos algoritmos dos SLRDA de mapeamento e exploração, em [14] foram estudados modelos de cadeias de SLRD, associados aos correspondentes SLRDA. Para tais modelos, esses algoritmos envolvem técnicas de inferência de SLRD, a fim de que os mesmos se auto ajustem, levando em consideração a similaridade em padrões repetitivos de segmentos. Estudos de inferência gramatical pela adaptatividade foram apresentados em [28] e [29].

IX. CONSIDERAÇÕES FINAIS.

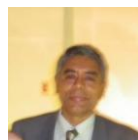
A despeito da relevância da representação computacional de SLRD, inclusive sendo uma área ativa de pesquisas há quase meio século conforme os levantamentos de [30] e [31], esse assunto ainda não foi explorado em navegação robótica, considerando cenários sujeitos a influências espúrias, com os recursos adaptativos listados em [32]. Esta pesquisa preenche tal lacuna, incorporando os fundamentos da geometria discreta aritmética de [33] ao método sintático em navegação robótica, por meio de técnicas adaptativas.

Este trabalho mostrou que o formalismo apresentado em [14] traz novas possibilidade à navegação robótica, agregando duas das principais vantagens verificadas na proposta de SLRDA: i) Simplicidade de modelagem e relativa facilidade de implementação, associadas a alto poder computacional; ii) Modelos fáceis de entender, relativamente simples de programar e flexíveis para aceitar mudanças em seu comportamento.

REFERÊNCIAS

- [1] Guoyu Zuo, Peng Zhang, and Junfei Qiao, "Path planning algorithm based on sub-region for agricultural robot," in Anais... march 2010, vol. 2, pp. 197 –200, 2nd International Asia Conference on Informatics in Control, Automation and Robotics (CAR).
- [2] N. Sariff and N. Buniyamin, "An overview of autonomous mobile robot path planning algorithms," in Anais... 4th Student Conference on Research and Development, June 2006.
- [3] Yeong-Hwa Chang, Tsu-Tian Lee, and Chang-Huan Liu, "On-line approximate cartesian path trajectory planning for robotic manipulators," IEEE Transactions on Systems, Man, and Cybernetics, vol. 22, no. 3, may 1992.
- [4] Sonja Macfarlane and Elizabeth A. Croft, "Jerk-bounded manipulator trajectory planning: Design for real-time applications," IEEE Transactions on Robotics, vol. 19, no. 1, pp. 42–52, February 2003.
- [5] Dhananjay Bodhale and Nitin Afzulpurkar, "Path planning for a mobile robot in a dynamic environment," in Anais... IEEE International Conference on Robotics and Biomimetics, 2008.
- [6] J. Sellen, "Direction weighted shortest path planning," in Anais... may 1995, vol. 2, pp. 1970 –1975 vol.2, IEEE International Conference on Robotics and Automation.
- [7] Stefan K. Gehrig and Fridtjof J. Stein, "A trajectory-based approach for the lateral control of car following systems," in Anais..., 1998, vol. 4, pp. 3596 – 3601.
- [8] B. H. Lee, "Algorithmic approach to straight line trajectory planning for mechanical manipulators," in Anais..., June 1986, pp. 121 – 126.

- [9] F. Schramm, A. Micaelli, and G. Morel, "Calibration free path planning for visual servoing yielding straight line behaviour both in image and work space," in *Anais.. aug. 2005*, pp. 2216 – 2221, International Conference on Intelligent Robots and Systems, (IROS 2005).
- [10] Rudiger Befit, Dietrich Paulus, and Michael Harbeck, "Segmentation of lines and arcs and its application for depth recovery," in *Anais... april 1997*, vol. 4, pp. 3165 –3168 vol.4, IEEE International Conference on Acoustics, Speech, and Signal Processing, ICASSP-97.
- [11] C.G. Cassandras, "Rapid learning techniques for discrete event systems," in *Anais.. jun 1993*, pp. 7/1 –7/8, IEE Colloquium on Discrete Event Systems: A New Challenge for Intelligent Control System.
- [12] A. H. R. Costa, *Robótica móvel inteligente: progressos e desafios*, Tese de livre docência, Escola Politécnica da Universidade de São Paulo, 2003.
- [13] Eliana P. L. Aude, Ernesto P. Lopes, Cristiano S. Aguiar, and Mario F. Martins, "Door crossing and state identification using robotic vision," in *Anais... 2006*, 8th International IFAC Symposium on Robot Control.
- [14] Leoncio Claro de Barros Neto, *Modelagem em geometria digital aprimorada por técnicas adaptativas de segmentos de retas*, Ph.D. thesis, Escola Politécnica da Universidade de São Paulo (USP), Junho 2011.
- [15] Leoncio C. de Barros Neto, André R. Hirakawa, and Antonio M. A. Massola, "An adaptive model applied to digital geometry to enhance segment straightness," *IEEE Latin America Transactions*, vol. 9, pp. 956 – 962, Oct. 2011.
- [16] H. Freeman, "Boundary encoding and processing," *Picture Processing and Psychopictorics*, pp. 241–266, 1970,
- [17] Azriel Rosenfeld, "Digital straight line segments," *IEEE Transactions on Computers*, vol. C-23, no. 12, pp. 1264–1269, December 1974.
- [18] Peter F.M. Nacken, "Metric for line segments," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 15, no. 12, pp. 1312–1318, December 1993.
- [19] F. Feschet, "The lattice width and quasi-straightness in digital spaces," in *Anais....*, Tampa, FL, December 2008, International conference on Pattern Recognition - ICPR, pp. 1–4.
- [20] Shu Xiang Li and Murray H. Loew, "Analysis and modeling of digitized straight-line segments," in *Anais....*, Rome, Italy, 1988, *Proceedings of International Conference on Pattern Recognition*, pp. 294–296, Publ by IEEE, Piscataway, NJ. S.M.
- [21] S.M. Aghito and F.S. Forchhammer, "Context-based coding of bilevel images enhanced by digital straight line analysis," *IEEE Transactions on Image Processing*, vol. 15, no. 8, pp. 2120–2130, August 2006.
- [22] D. Proffitt and D. Rosen, "Metrication errors and coding efficiency of chain-encoding schemes for the representation of lines and edges," *Computer Graphics and Image Processing*, vol. 10, no. 4, pp. 318–332, 1979.
- [23] Kai Ching You and King Sun Fu, "A syntactic approach to shape recognition using attributed grammars," *IEEE transactions on systems, man, and cybernetics*, vol. 9, no. 6, pp. 334–345, June 1979.
- [24] B. Garau, A. Alvarez, and G. Oliver, "Path planning of autonomous underwater vehicles in current fields with complex spatial variability: an A* approach," in *Anais.. april 2005*, pp. 194 – 198, IEEE International Conference on Robotics and Automation, ICRA 2005.
- [25] M. Tarokh, "A genetic robot path planner with fuzzy logic adaptation," in *Anais... july 2007*, pp. 388 –393, 6th IEEE/ACIS International Conference on Computer and Information Science.
- [26] M. A. A. Sousa, "Mapeamento de ambientes desconhecidos por robôs móveis utilizando autômatos adaptativos," M.S. thesis, Escola Politécnica da Universidade de São Paulo, 2006.
- [27] Leoncio C. de Barros Neto, André R. Hirakawa, and Antonio M. A. Massola, "Adaptive modeling of digital straightness applied to geometric representation enhancement," *International Journal of Computer Applications*, vol. 10, no. 2, pp. 31–39, November 2010.
- [28] Margaret Keiko Iwai, *Um formalismo gramatical adaptativo para linguagens dependentes de dependentes de contexto.*, Doutorado, Escola Politécnica da Universidade de São Paulo, São Paulo, 2000.
- [29] Ivone Penque Matsudo, "Um estudo dos processos de inferência de gramáticas regulares e livres de contexto baseados em modelos adaptativos,," Mestrado., Escola Politécnica da Universidade de São Paulo, 2006.
- [30] R. A. Klette and A. B. Rosenfeld, "Digital straightness: a review," *Discrete Applied Mathematics*, vol. 139, no. 1-3, pp. 197–230, April 2004.
- [31] Partha Bhowmick and Bhargab B. Bhattacharya, "Fast polygonal approximation of digital curves using relaxed straightness properties" *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 29, pp. 1590–1602, September 2007.
- [32] J. J. Neto, "Um levantamento da evolução da adaptatividade e da tecnologia adaptativa," *Revista IEEE América Latina*, vol. 5, no. 7, pp. 496–505, Novembro. 2007.
- [33] J. P. Reveillès, *Géométrie discrète, calcul en nombres entiers et algorithmique*, Ph.D. thesis, Université Louis Pasteur, Strasbourg, 1991.
- [34] Kenny Daniel et. al. A*: Any-Angle Path Planning on Grids. *Journal of Artificial Intelligence Research*, October 2010.
- [35] Kamil Tulum et. al. Situation Aware UAV Mission Route Planning. *IEEE Aerospace conference*, March 2009.
- [36] Ferdinand Wagner et. al. *Modeling Software with Finite State Machines: a Practical Approach*. Auerbach Publications, 2006.



Leoncio Claro de Barros Neto é formado em Engenharia Elétrica pela Escola de Engenharia Mauá, São Caetano do Sul, SP, em 1979. Concluiu o mestrado em 1994 e obteve o título de Doutor em Engenharia Elétrica em 2011, ambos pela POLI, USP. Em 1981 foi aprovado em concurso público de âmbito nacional, integrando o Corpo de Engenheiros Navais da Marinha do Brasil (MB). Na MB exerceu atividades técnicas em Setores de Manutenção e de Pesquisas, iniciando programa de doutorado na USP em 2006, após transferência para a reserva. Suas áreas de interesse e pesquisa abrangem automação, sensores inteligentes, eletrônica de potência, guerra eletrônica e suas aplicações.



André Riyuiti Hirakawa recebeu os títulos de Engenheiro Eletricista e correspondente mestrado pela USP em 1990 e 1992 respectivamente. Em 1992, como pesquisador, foi integrado no programa de doutorado da Yokohama National Universidade, em Yokiohoma, Japan, tendo recebido o título de PhD em 1997. Em 1998, iniciou atividades no Departamento de Engenharia de Computação e Sistemas Digitais da USP, sendo atualmente Professor Associado. Suas áreas de interesse e pesquisa abrangem automação e robótica, sensores inteligentes, eletrônica de potência, planejamento de rotas, AVG's, e sistemas sem fio, tanto aplicados à automação agrícola quanto em outros ambientes.