

1 Introdução

Em 2020, o mundo presenciou um acontecimento curioso: o bloqueio do Canal de Suez. Uma das mais importantes rotas comerciais marítimas havia sido bloqueada por um navio, causando um impacto global. Este acontecimento evidencia diversas fragilidades no sistema de comércio marítimo atual, mas também trás a tona um problema antigo, mas atual: a dificuldade na locomoção de objetos que navegam sobre ou abaixo da água visto sua inércia e os efeitos hidrodinâmicos existentes.

Na maior parte dos portos, para fazer a execução de manobras complicadas em espaços estreitos, utilizam-se rebocadores – barcos pequenos, mas que têm como característica principal sua agilidade e sua força. Em conjunto, esses barcos são usados para poder locomover grandes embarcações (Figura 1) e realizar tarefas que seriam impossíveis sem a ajuda deles.



Figura 1: Dois rebocadores manobrando um navio

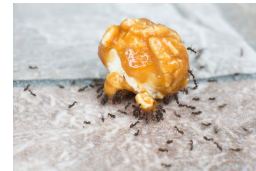


Figura 2: Formigas carregando uma pipoca

Na Biologia, pode-se enxergar um paralelo: formigas trabalhando em equipe para transportar alimentos de grande porte para o formigueiro (Figura 2), juntas realizam manobras complexas e conseguem transportar massas incrivelmente superiores quando comparadas às suas. Entretanto, nas manobras náuticas, utiliza-se quando muito três rebocadores, já as formigas, aproveitam de sua larga comunidade para realizar essas ações. Propõe-se o seguinte estudo: a análise de manobras náuticas usando uma "correção" de pequenos barcos.

2 Revisão bibliográfica

Um problema como este, pode ser abordado por diferentes frentes: pensando nele como um problema de controle, como um problema de otimização ou até mesmo como um problema de aprendizado de máquina.

Tentando atacá-lo como um problema de otimização, Braganza, Feemster e Dawson primeiramente tentaram fazer o desenvolvimento de um controlador de posição adaptativo que compensa os parâmetros selecionados do sistema, como coeficientes de massa e arrasto. O projeto de controle proposto é simplificado pela colocação estratégica dos rebocadores sobre o casco da embarcação, de modo que não exija a transmissão de informações entre os rebocadores [1].

Buscando encontrar uma solução menos suscetível a falhas em situações de mares fortes, o trabalho publicado por Esposito, Feemster e Smith [2] sugere usar uma otimização baseada em alocação de torque/força. Esposito continua esse trabalho criando um método para poder localizar uma posição ótima de novos rebocadores ao conjunto de forma a otimizar o aperto do conjunto (para melhorar a manipulação) [3].

As soluções propostas, entretanto, não permitem que os rebocadores se reposicionem, elas focam principalmente em calcular posições estratégicas e após isso controlar a embarcação maior. Essa é uma característica que não é vista quando observamos sistemas biológicos, estes se readaptam conforme a necessidade para se adequar a cada instante.

No estudo de controle de enxames robóticos, diversas técnicas inspiradas em comportamentos biológicos foram criadas, vê-se o uso da técnica de *Gradient-based Foraging* que simula estímulos como o de

feromônios[4]. Deseja-se neste trabalho, utilizar desta técnica para controlar robôs aquáticos que possam movimentar grandes massas.

3 Definição do problema

Para simplificar o problema, vamos supor que tanto o objeto que será movimentado, como os rebocadores são circulares, esta simplificação diminui em um a quantidade de graus de liberdade de cada objeto em questão, dado que a orientação não será relevante. Suponha n rebocadores robóticos, cada um posicionado em: $R_i : (x_i, y_i)$, com i variando de 1 a n . A posição do iceberg é dada por $S : (x, y)$, que deseja ser controlada. Tem-se as posições iniciais de cada rebocador, R_i^{start} , a posição do objeto a ser movimentado, S^{start} e a posição desejada do objeto, S^{goal} . O objeto começa em repouso e termina em repouso.

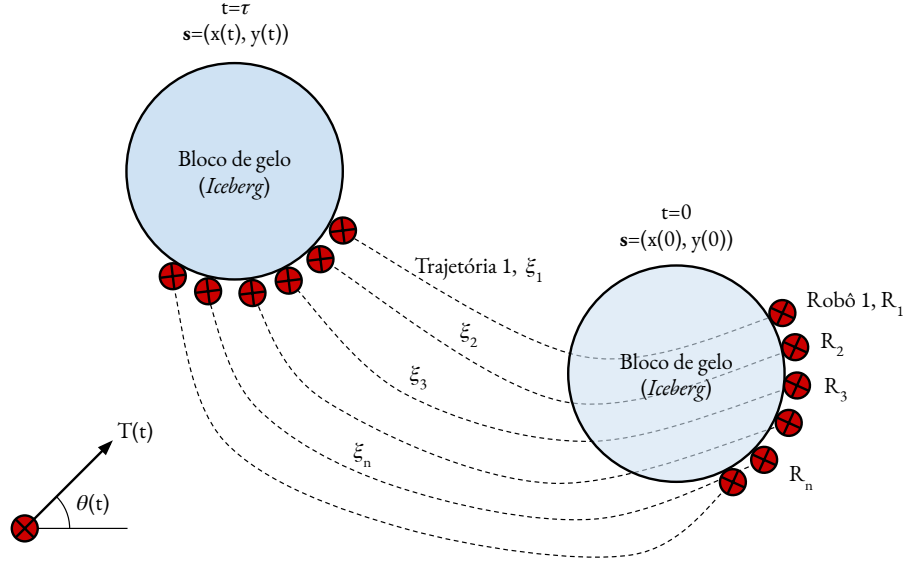


Figura 3: Esquema representativo do problema

Na Figura 3 observamos um exemplo de manobra a ser feita. Dadas as condições, deseja-se descobrir qual a velocidade necessária a ser produzida a cada instante pelos robôs, $v_i(t)$, para alcançar a posição desejada. Os robôs rebocadores têm ciência da posição deles próprios e de todos os outros, além de saber a posição do objeto a ser movimentado. O objeto tem diâmetro D e os rebocadores têm diâmetro d . Admite-se que não se tem incertezas na posição, e que o controle da velocidade de cada rebocador é ideal. Desconsideram-se atritos de onda, considera-se apenas o atrito de forma: $F_D \propto Av^2$,

	Variável	Complexidade Espacial
Condições do problema	S^{start}	$\mathcal{O}(1)$
	S^{goal}	$\mathcal{O}(1)$
	R_i^{start}	$\mathcal{O}(n)$
Informações de entrada	$R_i(t)$	$\mathcal{O}(n)$
	$\dot{R}_i(t)$	$\mathcal{O}(n)$
	$\ddot{R}_i(t)$	$\mathcal{O}(n)$
	$S(t)$	$\mathcal{O}(1)$
	$\dot{S}(t)$	$\mathcal{O}(1)$
Informações de saída	$\ddot{S}(t)$	$\mathcal{O}(1)$
	$v_i(t)$	$\mathcal{O}(n)$

Tabela 1: Complexidade espacial de algumas medidas do problema

3.1 Complexidade espacial (em notação *big O*)

A complexidade espacial representa a memória ocupada por um algoritmo para guardar a informação. De maneira semelhante à complexidade temporal, ela pode ser medida de forma assintótica pela notação *big O*, medindo a influência de um parâmetro na complexidade (no nosso caso, ligado à quantidade de barcos rebocadores), como pode-se ver na Tabela 1.

4 Interações

Na análise de um problema complexo, um fator relevante para se estudar são as interações entre os diferentes atores/agentes do sistema, uma interação se dá quando um agente afeta o estado de outro. O problema abordado apresenta dois tipos de atores: rebocadores e iceberg, que interagem entre si e com a água. Como visto anteriormente, os estados de cada um desses atores são definidos por uma posição e uma velocidade no plano bidimensional. Sendo assim, pode-se definir para este problema dois tipos de interações: as físicas e as informacionais.

4.1 Interações físicas

Definem-se interações físicas como interações causadas por forças. Tanto os rebocadores quanto o iceberg interagem da mesma forma com a água: ambos perdem energia por meio de atrito. Além disso, tanto os rebocadores quanto o iceberg interagem entre si por meio do choque mecânico.

4.1.1 Atrito

Como modo de simplificação do problema, pode-se considerar que os efeitos de atrito causados pela presença dos objetos na superfície da água sejam causados pelo arrasto, sendo assim são proporcionais ao quadrado da velocidade do objeto e com direção oposta à ela. A força de arrasto é definida da seguinte forma:

$$\vec{F}_D = -C_x \frac{\rho \vec{v} v}{2} A$$

Onde ρ é a massa específica da água, C_x o coeficiente de arrasto e A a área de referência. O casco dos robôs serão aproximados a esferas, neste caso, C_x é conhecido e vale 0.47. A área A será calculada a partir das dimensões do objeto.

4.1.2 Choque

Consideraremos que dois corpos interagem por meio de um choque/colisão quando se tocarem. Como todos os corpos do problema são circulares, dois rebocadores i e j interagem quando: $||\vec{R}_i - \vec{R}_j|| = d$ e um rebocador i interage com o iceberg quando $||\vec{R}_i - \vec{S}|| = \frac{d+D}{2}$. Pelo princípio da impenetrabilidade, consideramos: $||\vec{R}_i - \vec{R}_j|| \geq d$ e $||\vec{R}_i - \vec{S}|| \geq \frac{d+D}{2}$ sempre.

Assume-se que os choques são inelásticos, mas que se conserva a quantidade de movimento, ou seja:

$$\frac{d}{dt} \sum \vec{p}_i = 0 \text{ onde } \vec{p}_i \text{ é a quantidade de movimento do corpo } i$$

4.1.3 Influência das interações mecânicas

Como cada corpo estará sujeito a forças e impulsos causados pelos outros corpos, a influência destas forças e impulsos nos estados de um corpo seguem as leis de Newton:

$$\frac{d\vec{p}_i}{dt} = \vec{F}_i^{ext}, \text{ onde } \vec{p}_i = m_i \vec{v}_i$$

As posições são afetadas pela velocidade, dado que $\frac{d\vec{R}_i}{dt} = \vec{v}_i$.

4.2 Interações informacionais

Interações informacionais são aquelas causadas pela transmissão de informação. Na análise deste problema, assume-se que todos os robôs rebocadores sabem seu estado, o estado dos outros robôs e o estado do iceberg. O iceberg, por ser um agente passivo, não é influenciado pela informação dos estados dos outros agentes.

Entretanto, como os robôs rebocadores tomam decisões que podem afetar seu próprio estado de acordo com a informação que recebem dos outros ao seu redor, consideramos a transferência de informação como uma interação.

5 Experimento

Para resolver este problema, um ambiente virtual de simulação foi criado, este ambiente permite simular as leis mecânicas envolvidas neste problema, como choques e forças. O simulador foi construído em Python, usando as bibliotecas Pymunk e Pyglet.

Para controlar a velocidade de cada robô barco, três regras foram definidas, estas inspiradas pelo experimento de Craig Reynolds em 1986 sobre o comportamento de pássaros e peixes em conjunto, conhecido como *boid behaviour*.

5.1 Leis de comportamento

5.1.1 Lei da atração

Cada robô é atraído à superfície do iceberg seguindo a seguinte lei, com velocidade proporcional à distância da superfície:

$$\vec{v}_{a_i} = K_a * \left(\|\vec{S} - \vec{R}_i\| - \frac{D+d}{2} \right) \vec{u} \text{ onde } \vec{u} = \frac{\vec{S} - \vec{R}_i}{\|\vec{S} - \vec{R}_i\|}$$

5.1.2 Lei da repulsão

Os robôs, quando estão próximos repelem-se entre si com velocidade inversa à distância:

$$\vec{v}_{r_{ij}} = \begin{cases} K_r * \frac{1}{\|\vec{R}_j - \vec{R}_i\|} \vec{u} & \text{onde } \vec{u} = \frac{\vec{R}_j - \vec{R}_i}{\|\vec{R}_j - \vec{R}_i\|} \text{ caso } \|\vec{R}_j - \vec{R}_i\| < R_{min} \\ 0 & \text{senão} \end{cases}$$

5.1.3 Lei da direção

Nesta lei, usa-se de inspiração, o controle de velocidade por meio dos pontos de atração da técnica de controle por Campos de Potencial Virtual. Define-se a velocidade desejada do iceberg como:

$$\vec{v}_{iceberg_{des}} = K_d * (\vec{S}_{des} - \vec{S})$$

Projeta-se essa velocidade em todos os robôs nos quais $\vec{v}_{iceberg_{des}} \otimes (\vec{S} - \vec{R}_i) > 0$.

Esta condição permite que apenas robôs que estejam numa posição favorável a empurrar o iceberg sejam conduzidos a fazê-lo.

6 Resultados

O programa foi feito de tal modo que arquivos de configuração pudessem ser facilmente cambiados para poder mudar ganhos, posições iniciais, massas dos objetos e suas dimensões.

Dois arquivos de configuração foram criados: `config.yml` e `path.yml`.

O arquivo `path.yml` define os pontos da trajetória do iceberg:

```
1 points:
2   - x: 300
3     y: 300
4   - x: 100
5     y: 100
6   - x: 500
7     y: 100
8   - x: 500
9     y: 500
10  - x: 100
11    y: 500
```

Já o `config.yml` define as condições do ambiente de simulação:

```
1 world:
2   friction: 0.3
3   drag: 15
4   angular_drag: 150
5   k_a: 1.0
6   min_radius: 50
7   k_r: 100
8   k_d: 0.5
9 iceberg:
10  mass: 10
11  radius: 50
12  position:
13    x: 300
14    y: 300
15 robots:
16  mass: 1
17  radius: 10
18  positions:
19    - x: 100
20      y: 100
21    - x: 100
22      y: 200
23    - x: 100
24      y: 300
25    - x: 100
26      y: 400
27    - x: 100
28      y: 500
29    - x: 500
30      y: 100
31    - x: 500
32      y: 200
33    - x: 500
34      y: 300
35    - x: 500
36      y: 400
37    - x: 500
38      y: 500
```

Em um primeiro momento, calibraram-se os ganhos para obter resultados satisfatórios. Para as configurações acima, simula-se o comportamento dos robôs e do iceberg.

Na Figura 4 podem-se ver nove quadros da simulação, em que os robôs levam o iceberg para a posição correta desejada.

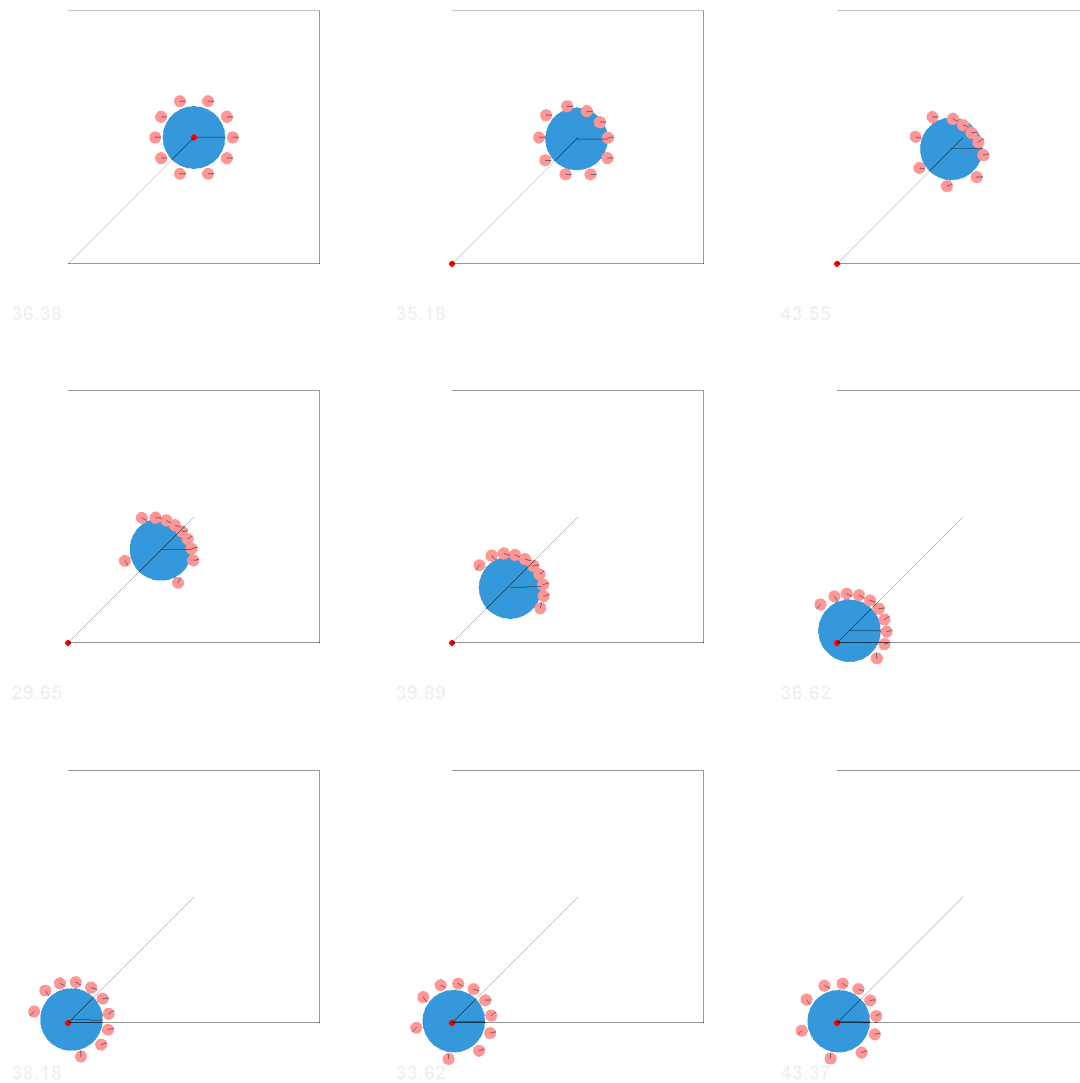


Figura 4: Simulação do deslocamento de um iceberg

O erro da posição do iceberg, mesmo que sendo um objeto passivo (que não apresenta um controle direto, sendo sujeito apenas a interações com os robôs) apresenta um comportamento desejável (tende a zero).

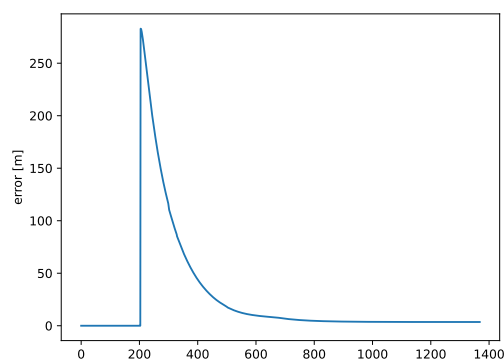


Figura 5: Evolução do erro da posição do iceberg

Um vídeo da simulação completa pode ser visto em: <https://media.giphy.com/media/bYDjzwml yjFfT7ZVMw/giphy.gif>.

7 Conclusões

O conjunto dessas três leis construídas permite que o controle da posição do iceberg seja muito robusta. Isso ocorre pois, os robôs sempre querem permanecer próximos ao iceberg, mas nunca tão próximos uns dos outros, fazendo com que eles se dispersem no contorno do iceberg. Quando o iceberg está longe da posição desejada, entretanto, o efeito da dispersão torna-se pouco importante, e então, os robôs se unem para levar o objeto à posição desejada. Quando chegam próximos a ela, a velocidade imposta pelo controle de direção fica pequena novamente, e eles voltam a se dispersar, permitindo se readaptar às novas possíveis posições desejadas.

A solução construída, apesar de resolver o problema definido, depende do fato que o controle por velocidade seja perfeito. Não foram simulados erros de controle. Como continuação deste trabalho, gostaria de poder realizar ensaios experimentais para observar se a solução proposta poderia funcionar em um ambiente real.

Referências

- [1] D. Braganza, M. Feemster and D. Dawson, "Positioning of Large Surface Vessels using Multiple Tugboats," 2007 American Control Conference, 2007, pp. 912-917.
- [2] Esposito, J.; Feemster, M.; Smith, E. Cooperative manipulation on the water using a swarm of autonomous tugboats. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.
- [3] Esposito, J. Distributed grasp synthesis for swarm manipulation with applications to autonomous tugboats. In Proceedings of the 2008 IEEE International Conference on Robotics and Automation, Pasadena, CA, USA, 19–23 May 2008.
- [4] Nicholas Hoff. 2011. "Multi-Robot Foraging for Swarms of Simple Robots." Doctoral Thesis, Harvard University. Thesis Type: phd.