Modelagem Inicial do Guindaste usando CoppeliaSim

João Victor de Castro Alves,
Juliana Assis Alves
Pierre Victor da Silva Sousa
Roberto Pires Oliveira
Vítor Corrêa Silva

GRUPO D

1 Introdução

Nesse trabalho iremos modelar a versão básica do guindaste com leves modificações no software CoppeliaSim. Isso tem como objetivo não só comprovar o entendimento do grupo sobre o software e a tarefa mas também consolidar os conhecimentos adquiridos tendo em vista que o projeto modelado deverá ser implementado em sua forma física.

2 Definição do Problema

O primeiro passo para desenvolver o modelo é estabelecer quais são as diretivas que iram guiar nosso design. Tendo em vista que nenhuma restrição de dimensão ou design foi diretamente feita pelo professor, iremos usar somente as carcteristicas do problema que o modelo deve solucionar.

A definição formal do nosso problema pode ser escrita como :

"O guindaste deve ser capaz de pegar uma moeda de 50 centavos a uma altura de 7mm e 150mm de distância do guindaste levar essa moeda até uma base, também a 150mm de distância e com uma altura de até 220mm."

3 Modelagem das peças

Utilizando somente essas definições é possível já extrair algumas das medidas físicas que nosso guindaste deve ter. Permitindo assim a criação de um modelo suficientemente satisfatório. Sendo assim é possível escrever essas definições da seguinte forma :

- A altura da torre deve exceder $220mm$.

- O comprimento da lança deve ser, no mínimo, 150mm.
- O atuador deve ser capaz de acoplar e manter acoplado por tempo indeterminado um objeto metálico de 8g.

3.1 Base

A base é responsável por garantir a estabilidade de toda a estrutura do guindaste. Temos que garantir que a base tenha tamanho e peso suficiente para garantir que o guindaste não caia, estando ele carregado ou não. Portanto a peça que será usada terá as seguintes caractéristicas :

- Tipo : Cuboid
- Dimensões : 70mm x 70mm x 20mm
- Massa : 200g

3.2 Torre

A Torre precisa atender a restrição de altura tendo uma nível razoável de ”folga”, e ainda sendo factível que esta torre consiga carregar a Lança estando ou não em movimento. Portanto a peça que será usada terá as seguintes caractéristicas :

- Tipo : Cuboid
- Dimensões : 50mm x 50mm x 300mm
- Massa : 450g

3.3 Lança

A lança é responsável por mover atuador até a posição alvo e deve ser capaz de aguentar tanto o peso da carga quanto da estrutura do atuador

- Tipo : Cuboid
- Dimensões : 250mm x 30mm x 15mm
- Massa : 200g

3.4 Contrapeso

O contrapeso tem a função de evitar o desequilíbrio da Lança e garantir que o guindaste não tombe mesmo com o peso da carga.

- Tipo : Cuboid
- Dimensões : 30mm x 30mm x 30mm
- Massa : 120g

3.5 Atuador

O atuador usado será feito de forma customizada no software de forma que sua geometria é de escolha livre. A geometria escolhida será a de um pequeno cilindro, buscando mimetizar a peça utilizada na versão física a ser construída. Portanto :

- Tipo : Cylinder
- Dimensões : $30mm$ x $30mm$ x $60mm$
- Massa : $300g$

3.6 Base do Atuador

Como o atuador será movido tbm horizontamente ele precisará de uma base própria que terá as seguintes caracteristicas:

- Tipo : Cuboid
- Dimensões : $30mm$ x $30mm$ x $30mm$
- Massa : $30g$

4 Sensoriamento

Para ser possível ter o set completo de funcionalidades visando controle estável do guindaste será adicionado sensoriamento especificamento um tipo de sensor:

4.1 Sensor de Distância

Será usado um sensor de distância, especificamente do tipo raio, para calcular a distância entre o atuador e tudo que está logo abaixo. As caracteristicas de configuração podem ser vistas abaixo :

- Tipo : Ray
- OffSet = $5mm$
- Distância Máxima = $250mm$

5 Juntas

Juntas são objetos que tem o a função de simular movimento de uma estrutura em relação a outra. Esse movimento foi pode ter múltiplos parâmetros definidos porém para a essa iteração da simulação usaremos somente dois. O módulo máximo de velocidade e o torque máximo.

5.1 Suporte

A junta de suporte tem como objetivo garantir a união da torre a base gerando assim estabilidade para o guindaste. Sua velocidade estará definida como zero e seu torque terá um valor simbólico de 100 N, buscando dar rubustez ao modelo:

- $\left|\vec{V_{max}}\right| = 0$
- $T_{max} = 200$

5.2 Plano XY

- $\left|\vec{V_{max}}\right| = 20°/S$
- $T_{max} = 200$

5.3 Eixo Z

- $\left|\vec{V_{max}}\right| = 6 \cdot 10^{-2} m/S$
- $T_{max} = 100$

5.4 Alinhamento Horizontal

Assim como no vídeo, o guindaste em questão também tera três graus de liberdade portanto é necessário uma junta prismática para gerar esse movimento. Ela terá as mesmas limitações da junta do Eixo Z.

- $\left|\vec{V_{max}}\right| = 6 \cdot 10^{-2} m/S$
- $T_{max} = 100$

6 API

Todo o controle do modelo será feito via API usando a linguagem Python. Para isso as seguintes classes e métodos foram desenvolvidos.

6.1 Template

Para ser possível iteragir com todas as estruturas de forma padrão usando o handle será criada a classe template object, que irá conter além do handle do objeto a que se refere, o canal de comunicação para a simulção. Sendo esse um parâmetro compartilhado entre múltiplas instâncias. Todas as classe que foram definidas a seguir tendo essa classe ou uma classe filha como parente.

6.2 Juntas

Para esse modelo a funcionalidade de Control Loop será usada na junta. É uma modalidade aonde a velocidade máxima da junta é definida, e só a sua posição é controlada via API. Para isso foram definidos dois métodos :

- getPosition : Retorna a posição da junta em relação aos seus limites superior e inferior.
- setPosition : Define a posição objetivo para a junta em relação aos seu limite superior e inferior

6.2.1 Junta de Revolução

A gente de revolução não apresenta nenhum método novo porém tem os métodos anteriores modificados para trabalharem com dados em radianos e em graus.

6.2.2 Junta Prismática

Não há nenhuma modificação em relação a classe mãe 'Juntas', a existência dessa classe é puramente para future-proofing e clareza de código.

6.3 Sensor de Distância

A classe do sensor de distância é derivada da classe 'Objeto' e possui o método necessário para ler e interpretar as leituras do sensor de distância.

- detect : Retorna o handle, assim como a distânca do objeto detectado. Caso não encontre um objeto retorna o valor infinito.

7 Atuador

O atuador é por sí só um modelo autoral que necessita uma seção própria. Seu modelo consiste em três peças:

- Corpo : Consiste em un cilíndro de dimensões já explicitadas em sessões anteriores e tem a função de representar o corpo do imã que será usado na vida real.
- Sensor de Força : Será usada simplesmente como um link entre o corpo e o objeto alvo, como nenhuma de suas funcionalidade está sendo utlizada não se mostrou necessário descrever seu funcionamento.
- Sensor de Distância : O mesmo tipo de sensor usado para medir a distância entre a ferramenta e o chão, entretanto esse apresenta um range de somente $20mm$.

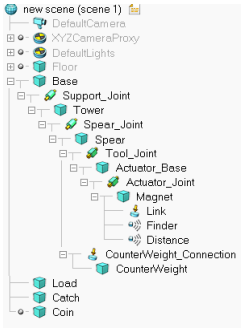
7.1 Comportamento

A proposta para o modelo de imã implementado é que utilizando o sensor de distância para encontrar o objeto no alcance de magnetização, o mesmo seja linkado a árvore do sensor de força estabelecendo a ligação entre ele e o corpo do modelo e permitindo movimenta-lo até o destino e deposita-lo lá.

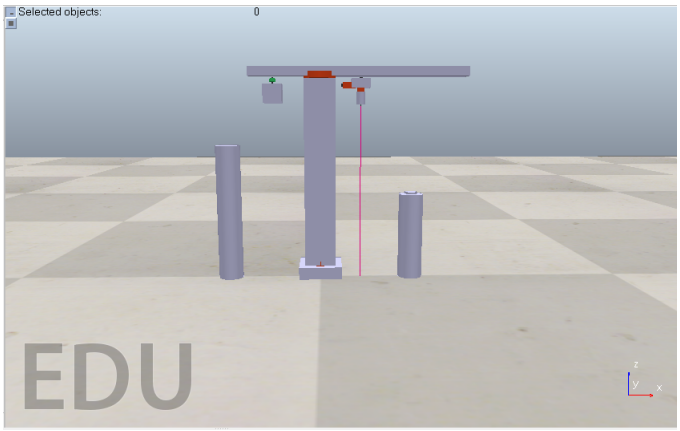
Para permitir essa modelagem foram usadas funções da API Regular que retorna afetam diretamente a árvore de hierárquia de cena. Essas funções permitem que possamos fazer a inserção e retirada do objeto desejado da ramo desejado da árvore, durante a execução da simulação. Não hávera muitos detalhes do código implementado pois não se mostra necessário para esse relatório.

8 Resultados

Nesta sessão seram exibidos os resultados da modelagem, primeiramente, a árvore de hierarquia de cena final do modelo :

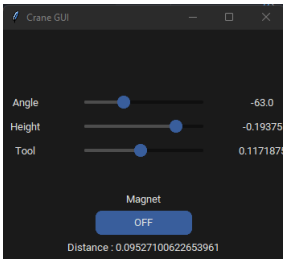


O resultado final do modelo pode ser visto na imagem a seguir :



Para ser possível controlar esse modelo de forma intuitiva e prática uma GUI

básica foi programada usando o TKinter, uma biblioteca de desenvolvimento de interfaces em python.



Para ser possível ver o modelo em pleno funcionamento é necessário acesssar o seguinte link `https://youtu.be/rzIPiQRTQPA`

Referências

- COPPELIA REGULAR API, Dísponivel em : `https://www.coppeliarobotics.com/helpFiles/en/apiFunctions.htm`. Acessado em 2 de Maio de 2022
- V-REP Beginners Tutorial, Dísponivel em `https://www.youtube.com/playlist?list=PLxtIrkqXXpsg2OiTdisdZ59rj_47a886x`. Acessado em 2 de Maio de 2022