FACULDADE DE CIÊNCIAS E TECNOLOGIA
UNIVERSIDADE D COIMBRA

Fundamentos de Inteligência Artificial

Licenciatura em Engenharia Informática

2025

Relatório | TP 1 – Lunar Lander

Projeto realizado por:

André Miguel Correia Cardoso | 2022222265 - uc2022222265@student.uc.pt - PL2

João Filipe Valente Cardoso | 2022222301 - uc2022222301@student.uc.pt - PL2

Sérgio Lopes Marques | 2022222096 - uc2022222096@student.uc.pt - PL2

Após efetuada a análise ao enunciado e compreendido o objetivo do trabalho, definimos as perceções e as ações do agente reativo em estudo, com base nas informações que nos foram fornecidas.

Perceções

Foi possível perceber que a nossa nave recebe 8 perceções do ambiente:

- x - Posição horizontal da nave (0 = centro da plataforma)
- y - Altitude da nave
- vx - Velocidade horizontal
- vy - Velocidade vertical (valores negativos indicam queda)
- theta - Ângulo de inclinação da nave
- vtheta - Velocidade angular (rotação)
- left_leg - Estado da perna esquerda (1 = em contacto com o solo)
- right_leg - Estado da perna direita (1 = em contacto com o solo)

No código python desenvolvido as perceções do agente foram definidas dentro da função “perceptions()”, que retorna um dicionário que contém o valor de cada uma das perceções observado num determinado instante da simulação.

```python
def perceptions(observation):
    return {
        "x": observation[0],
        "y": observation[1],
        "vx": observation[2],
        "vy": observation[3],
        "theta": observation[4],
        "vtheta": observation[5],
        "left_leg": observation[6],
        "right_leg": observation[7]
```

Figura 1 – *Função perceptions()*

Este dicionário é, posteriormente, utilizado na função “actions()”, onde está definido o nosso sistema de produções, que permite ao nosso agente tomar uma ação consoante a perceção observada.

Espaço de Ações

As ações da nossa nave espacial são efetuadas tendo em conta a utilização de 3 motores, que são controlados pelos valores de um vetor com dois números reais.

- action[0] - Motor principal: controla a força de propulsão vertical
- action[1] - Motores secundários: controlam a rotação da nave
    - Valores negativos: motor direito ligado -> rotação da nave no sentido anti-horário
    - Valores positivos: motor esquerdo ligado -> rotação da nave no sentido horário

Depois de algumas experiências com a utilização de diferentes abordagens, alterando os valores das perceções para determinadas ações, encontrámos um sistema de produções que reúne as condições desejadas, tendo um bom desempenho com e sem a presença de vento, com boas percentagens de aterragem sucedida para ambas as situações.

Sistema de Produções

De seguida, está descrito o sistema de produções desenvolvido, com base nas perceções e ações definidas:

abs(x > 0.1), vy < 0 --> action[0] = 1.0

vy < -0.4 --> action[0] = 1.0

left_leg or right_leg --> NIL

~left_leg, ~right_leg, vx > 0.1, 0 < theta < 0.1 --> action[1] = -1.0

~left_leg, ~right_leg, vx < -0.1, -0.1 < theta < 0 --> action[1] = 1.0

~left_leg, ~right_leg, vtheta > 0.1 --> action[1] = 1.0

~left_leg, ~right_leg, vtheta < -0.1 --> action[1] = -1.0

~left_leg, ~right_leg, x > 0.1, vx > 0, 0 < theta < 0.2 --> action[1] = -1.0

~left_leg, ~right_leg, x < -0.1, vx < 0, -0.2 < theta < 0 --> action[1] = 1.0

~left_leg, ~right_leg, theta > 0 --> action[1] = 1.0

~left_leg, ~right_leg, theta < 0 --> action[1] = -1.0

Regras para o Motor Principal (Propulsão Vertical):

- **se** a nave está afastada do centro (|x| > 0,1) **e** está em queda (vy < 0), **então** ativar o motor principal, uma vez que a plataforma de aterragem se encontra no ponto (0,0) e uma aterragem bem sucedida implica que a nave esteja em abs(x) <= 0,2. O valor 0,1 escolhido é mais restritivo, garantindo que a nave se aproxime de forma correta do centro da plataforma.
- **se** a velocidade de queda é muito alta (vy < -0,4), **então** ativar motor principal. Como podemos observar no código que nos foi fornecido, a velocidade para uma aterragem com sucesso (“stable_velocity”) deve ser maior que -0,2. O valor -0,4 é utilizado para evitar que a nave atinga velocidades verticais muito grandes e garantir que esta se encontra com uma velocidade com valor dentro dos valores pretendidos.

Estas regras visam:

1. Controlar a velocidade de descida quando a nave se encontra afastada do centro
2. Evitar velocidades verticais excessivas que poderiam levar a uma colisão destrutiva

Regras para os Motores Secundários (Rotação):

- **se** pelo menos uma das duas pernas da nave estiver em contacto com o solo, **então** não há necessidade de ligar qualquer motor secundário, com o intuito de fazer com que a nave estabilize e coloque a outra perna no solo.
- **se** a nave está a mover-se para a direita (vx > 0,1) **e** levemente inclinada para a direita (0 < theta < 0,1), **então** é ligado o motor direito, efetuando uma rotação anti-horária, para tentar estabilizar a nave.
- **se** a nave está a mover-se para a esquerda (vx < -0,1) **e** levemente inclinada para a esquerda (-0,1 < theta < 0), **então** é ligado o motor esquerdo, realizando-se uma rotação no sentido horário, com a intenção de estabilizar a nave.
- **se** a nave está a girar muito rapidamente no sentido anti-horário (vtheta > 0,1), **então** é efetuada uma rotação horária, ao ser ligado o motor esquerdo, com a tentativa de contrabalançar a nave.

- **se** a nave está a girar muito rapidamente no sentido horário (vtheta < -0,1), **então** efetua-se uma rotação anti-horária, ligando-se o motor direito, contrabalançando, assim, a nave.
- **se** a nave está à direita do centro (x > 0,1) **e** a mover-se para a esquerda (vx > 0) **e** inclinada para a esquerda (0 < theta < 0,2), **então** é ligado o motor direito, com o intuito de fazer a nave ir de encontro com o centro da plataforma.
- **se** a nave está à esquerda do centro (x < -0,1) **e** a mover-se para a direita (vx < 0) **e** inclinada para a direita (-0,2 < theta < 0), **então** é ligado o motor esquerdo, fazendo com que a nave se dirija para o centro da plataforma.
- **se** a nave está inclinada para a esquerda (theta > 0), **então** é efetuada uma rotação horária, ligando-se o motor esquerdo, de modo a corrigir a inclinação da nave, tentando estabilizá-la.
- **se** a nave está inclinada para a direita (theta < 0), **então** realiza-se uma rotação anti-horária, ativando-se o motor direito, numa tentativa de estabilizar a nave.

Os valores das perceções utilizados nas condições do nosso sistema de produções devem-se a informações retiradas do enunciado e do código fornecidos e às experiências realizadas durante o desenvolvimento do código:

theta > 0,2 (12 graus): Este valor é escolhido com base na informação retirada do código fornecido, que sugere uma orientação para aterragem co sucesso de 20º. Para restringir um pouco mais os movimentos da nave, impedindo que esta tenha oscilações muito acentuadas e, depois de algumas experiências com valores diferentes, foi escolhido o valor de 12º.

vtheta > 0,1 / **vx** > 0,1: Em relação à velocidade angular e à velocidade horizontal, não é recomendado nenhum valor na documentação fornecida. Contudo, escolhemos o valor 0,1 uma vez que manter valores baixos para estes parâmetros é essencial para alcançar uma orientação estável na aterragem.

Estas regras procuram:

1. Estabilizar a rotação da nave
2. Corrigir desvios de posição na horizontal
3. Contrariar velocidades horizontais excessivas

Como referido anteriormente, as regras do sistema de produções estão definidas no nosso código python na função “actions()”, através das condições “if” apresentadas de seguida:

```python
def actions(perceptions):

    action = [0, 0]

    if (abs(perceptions["x"]) > 0.1 and perceptions["vy"] < 0):
        action[0] = 1.0
    elif perceptions["vy"] < -0.4:
        action[0] = 1.0
    if perceptions["left_leg"] or perceptions["right_leg"]:
        pass
    else:
        if perceptions["vx"] > 0.1 and 0 < perceptions["theta"] < 0.1:
            action[1] = -1.0
        elif perceptions["vx"] < -0.1 and -0.1 < perceptions["theta"] < 0:
            action[1] = 1.0
        elif perceptions["vtheta"] > 0.1:
            action[1] = 1.0
        elif perceptions["vtheta"] < -0.1:
            action[1] = -1.0
        elif perceptions["x"] > 0.1 and perceptions["vx"] > 0 and 0 < perceptions["theta"] < 0.2:
            action[1] = -1.0
        elif perceptions["x"] < -0.1 and perceptions["vx"] < 0 and -0.2 < perceptions["theta"] < 0:
            action[1] = 1.0
        elif perceptions["theta"] > 0:
            action[1] = 1.0
        elif perceptions["theta"] < 0:
            action[1] = -1.0

    return action
```

Figura 2 – *Função actions()*

Análise de Performance e Resultados Obtidos

Durante o desenvolvimento do código, foi realizada uma análise contínua à performance da nossa nave espacial, com o objetivo de perceber se as alterações efetuadas iriam afetar a simulação de forma positiva.

Como a visualização de um conjunto de 5 ou 10 simulações não era a abordagem mais fiável, foram realizados testes com 1000 experiências diferentes, para determinar a taxa de sucesso de aterragem, com e sem a presença de vento.

Para a última e mais aperfeiçoada versão do nosso sistema de produções, obtivemos os seguintes resultados:

Sem Vento: Taxa de sucesso → 92,8%

Média de passos → 34321

Com Vento: Taxa de sucesso → 71.1%

Média de passos → 56273

Ao observar estes resultados, podemos perceber que a presença de vento lateral complica a ação do agente reativo.

A taxa de aterragens com sucesso baixa mais de 20%, quando comparada com a situação onde o vento não está presente na simulação. A média de passos das aterragens bem sucedidas para as duas situações indica, também, que, com a presença de vento, as aterragens foram mais demoradas, havendo uma diferença significativa de mais de 20000 passos.

Desafios Encontrados

Durante o desenvolvimento do agente, enfrentámos vários desafios, que fomos conseguindo resolver à medida que avançávamos nas experiências e nos testes realizados.

Um desses desafios já foi referido anteriormente: na situação onde os ventos laterais estão presentes, o agente reativo tinha dificuldade em aterrar sobre a plataforma. Por vezes, aterrava fora da zona delimitada pelas bandeiras e, noutras situações, era, até, arrastado para fora do espaço da simulação.

Outras situações onde demonstrámos ter mais dificuldades estavam relacionadas com a parte final da aterragem da nave. Em alguns casos, a nave aterrava dentro dos limites definidos pela bandeira, mas continuava em movimento, deslizando para fora dos limites. Por vezes, acontecia, também, que a nave não aterrava com as duas pernas exatamente ao mesmo tempo no solo, fazendo, portanto, com que a aterragem não tivesse sucesso.

Estes problemas não foram totalmente eliminados. Contudo, conseguimos mitigá-los ao fazer as alterações corretas aos valores utilizados no sistema de produções.