

Robótica e Sistemas Inteligentes

Relatório da unidade curricular apresentado no âmbito de provas de agregação

Nuno Miguel Fonseca Ferreira

Vila Real, março de 2019

Relatório da unidade curricular de Robótica e Sistemas Inteligentes apresentado no âmbito de Provas de Agregação, de harmonia com o disposto na alínea b), do número 2, do artigo 8º do Decreto-Lei nº 239/2007, de 19 de junho de 2007.

Índice

Lista de Acrónimos.....	iii
1. Introdução.....	1
2. Enquadramento do ensino de Robótica e Sistemas Inteligentes	2
3. Objetivos da unidade curricular.....	3
4. Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular	4
5. Demonstração da coerência das metodologias de ensino com os objetivos da unidade curricular	4
6. Conteúdo programático e planificação.....	5
7. Competências a desenvolver	6
8. Metodologia de ensino.....	7
9. Metodologia de avaliação.....	9
10. Repertório bibliográfico.....	11
11. A unidade curricular em outras instituições de ensino superior.....	13
12. Considerações Finais	16
Referências	17
Anexo A – Trabalhos práticos propostos.....	19
Anexo B – Guião do ROS	25
B1. Instalação do Sistema Operativo ROS	25
B2. Comunicação entre o ROS e o simulador Turtlesim	28
B3. Instalação do vídeo	30
B4. Instalação do Arduino	31
B5. Exemplos práticos.....	32
B6. Instalação do sistema multi-robôs no ROS.....	33
Anexo C – Exemplo de Exame de Avaliação	35
Anexo D – Plano curricular do MIEEC - UTAD.....	37

Lista de Acrónimos

A3ES	Agência de Avaliação e Acreditação do Ensino Superior
EEC	Engenharia Eletrotécnica e de Computadores
FEUP	Faculdade de Engenharia da Universidade do Porto
IA	Inteligência Artificial
ISEP	Instituto Superior de Engenharia do Porto
ISEC	Instituto Superior de Engenharia de Coimbra
MEEC	Mestrado em Engenharia Eletrotécnica e de Computadores
MIEEC	Mestrado Integrado em Engenharia Eletrotécnica e de Computadores
NF	Nota final
P	Prática Laboratorial
PE	Prova Escrita
RSI	Robótica e Sistemas Inteligentes
T	Teórica
TP	Teórico-prático
UC	Unidade Curricular
UTAD	Universidade de Trás-os-Montes e Alto Douro

1. Introdução

Este relatório, elaborado no âmbito da realização de Provas de Agregação, diz respeito à unidade curricular (UC) de Robótica e Sistemas Inteligentes (RSI). A UC escolhida não integra nenhum plano de estudos acreditado mas considerando o interesse e o enquadramento da área de conhecimentos em questão poderia fazer parte do plano de estudos de um mestrado em Engenharia Eletrotécnica e de Computadores (EEC) ou de um programa doutoral de EEC ou até de Engenharia Informática.

Esta UC foi organizada para 15 semanas letivas, com uma carga horária total de 30 horas de ensino teórico (T e TP) e 30 horas de ensino prático e laboratorial (P). Neste relatório apresenta-se a UC de RSI ao nível do conteúdo programático que poderá ser proposto para a Agência de Avaliação e Acreditação do Ensino Superior (A3ES).

A escolha desta UC recaiu no facto do autor, do presente relatório, ser Professor na área de Engenharia Eletrotécnica, bem como constituir uma área de interesse e enquadrar-se, no seu percurso de investigação e, além disso, estabelecer sinergias com outras UC, nas quais, colabora na lecionação, nomeadamente, na UC de Sistemas de Automação e Controlo ambas do curso de Mestrado em Engenharia Eletrotécnica e de Computadores (MEEC) no Instituto Superior de Engenharia de Coimbra (ISEC).

O presente relatório está organizado por secções e quatro anexos.

A secção 2 apresenta a importância e o enquadramento da UC, os objetivos gerais são descritos na secção 3 e nas secções 4 e 5 demonstra-se a coerência dos conteúdos programáticos e das metodologias de ensino com os objetivos da UC, na secção 6 é descrito o programa detalhado e o cronograma dos conteúdos da UC ao longo do semestre. Na secção 7 indicam-se os objetivos específicos que os alunos devem alcançar e as competências que devem desenvolver ao longo do semestre. As metodologias de ensino e de avaliação adotadas estão descritas respetivamente nas secções 8 e 9 e na secção 10, apresenta-se o repertório bibliográfico a que os discentes podem recorrer. Na secção 11, enquadra-se a UC de RSI aqui proposta, nos programas de UC similares de outras instituições de ensino superior nacionais e internacionais. Por último tecem-se algumas considerações finais e apresenta-se um conjunto de informação complementar, anexos A, B e C, que integram exemplos de trabalhos práticos, um guião de utilização e um teste

de avaliação e o anexo D contendo o plano curricular do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores (MIEEC) da Universidade de Trás-os-Montes e Alto Douro (UTAD).

2. Enquadramento do ensino de Robótica e Sistemas Inteligentes

Ao longo dos séculos foram sendo criadas ferramentas cada vez mais complexas. Ferramentas físicas tiveram um papel fundamental na história da humanidade. Paralelamente, o ser humano também desenvolveu técnicas de comunicação, como a linguagem oral, escrita, incluindo a matemática, que, para além de potenciarem a troca e armazenamento de informação, permitiram a expressão de conceitos através de outras perspetivas. A invenção do computador desencadeou uma revolução no processamento de um vasto conjunto de tarefas, desde trabalhos fabris repetitivos e fastidiosos até à utilização do banco eletrónico, que foram simplificadas e tornadas mais fiáveis.

Hoje em dia vive-se na era da revolução do conhecimento, na qual a vantagem tecnológica das nações se mede pelo conhecimento que detêm, impulsionado, sobretudo, pela ciência. No entanto, o sucesso só se alcança com profissionais altamente qualificados. Desta forma, a RSI desempenha um papel fundamental neste processo, permitindo aos discentes adquirir competências transversais em diversas áreas de conhecimento.

Assiste-se atualmente a um forte impulso da comunidade científica para aumentar significativamente a penetração na sociedade das tecnologias da robótica em geral e da robótica móvel em particular. Habitualmente designadas por robótica de serviços, através da utilização de robôs em espaços de trabalho dinâmicos e não estruturados partilhados com pessoas, por vezes em estreita interação e colaboração com pessoas, em diversos sectores de atividade e da sociedade, abrangendo locais de trabalho na área de serviços, espaços públicos e de lazer, ambientes domésticos, ambientes industriais e agricultura de precisão. A robótica possui uma presença inquestionável em diversos contextos, nomeadamente em ambientes industriais. Ao longo das últimas décadas, a robótica móvel tem recebido particular atenção por parte da comunidade científica e industrial, focando-se em aspetos essenciais para a adequada mobilidade e autonomia dos robôs em cenários reais, tais como o mapeamento e localização simultâneos, o planeamento de trajetórias, a navegação autónoma, a coordenação, a cooperação e comunicação de múltiplos robôs, a exploração autónoma, a vigilância, o reconhecimento de objetos, entre outros.

O esforço da comunidade científica para dotar os robôs móveis destas características tem vindo a beneficiar o progresso da inteligência artificial (IA), da teoria de controlo, dos sistemas embebidos, dos sistemas de comunicação, da integração de sensores e fusão de dados, da perceção artificial, e outras áreas associadas à robótica móvel.

Atualmente e de um modo geral, o desafio da IA consiste em desenvolver uma máquina capaz de raciocinar. A IA, juntamente com outros ramos da ciência, proporcionou o aparecimento de várias ferramentas que usam sistemas baseados em conhecimento utilizadas no controlo de sistemas robóticos.

A RSI permite manusear um conjunto vasto de problemas de engenharia e de praticamente todas as outras ciências, anteriormente intratáveis ou muito difíceis, admitindo, ainda, a abordagem de outros tipos de problemas de forma mais eficiente. Normalmente incorporam um computador que tenta resolver os processos de forma semelhante aos seres humanos. Em suma, a RSI têm vindo a desempenhar um papel relevante na sociedade, fazendo parte integrante da nossa vida diária. Neste sentido, torna-se essencial que os alunos do curso de MIEEC estejam preparados para lidar com diversos sistemas, que se adaptam de acordo com o ambiente e como as pessoas que interagem com eles. A UC de RSI visa preparar os alunos para o mercado de trabalho onde, no futuro, qualquer dispositivo terá um comportamento inteligente.

3. Objetivos da unidade curricular

A unidade curricular de RSI pretende oferecer aos discentes uma panorâmica dos temas e tecnologias envolvidas na conceção e aplicação de robôs em sistemas industriais em que os discentes verificarão a compreensão destes tópicos através da implementação de sistemas reais.

O objetivo principal é formar os discentes sobre os princípios da robótica e seus campos de aplicação. Na presente unidade curricular, pretende abordar-se de forma exaustiva e pormenorizada, a terminologia usada na robótica de modo a que obtenham o conhecimento que lhes permita a identificação de aplicações suscetíveis de robotizar.

Com os conhecimentos ministrados nesta UC pretende-se que os discentes possam compreender a fundo o essencial da robótica que compõem um dos processos da automação das indústrias, e que consigam facilmente aprender e compreender uma qualquer linguagem de programação de sistemas robóticos.

Apesar das aplicações da RSI ser vasta, o seu desenvolvimento assenta sobre um conjunto de técnicas básicas. Desta forma, constitui um propósito fundamental desta UC, numa área reconhecidamente importante para os setores da indústria e dos serviços, proporcionar ao aluno uma formação fundamental relativamente aos conceitos subjacentes à RSI.

4. Demonstração da coerência dos conteúdos programáticos com os objetivos da unidade curricular

Os objetivos da UC assentam na profunda compreensão dos princípios fundamentais do controlo e da robótica que compõem as diversas aplicações industriais. Assim, os conteúdos programáticos concebidos para a UC visam uma abrangência extremamente ampla desde sistemas sensoriais que compõem os sistemas robóticos envolvidos nas indústrias.

Os conteúdos da UC estão organizados por partes, de acordo com a tipologia dos assuntos a abordar, e no seu todo constituem uma perspetiva bastante abrangente das aplicações usadas em sistemas industriais. Nomeadamente, desde as aplicações mais clássicas como os sistemas mais complexos, até às tecnologias emergentes e mais recentes, focalizando a perspetiva globalizante e que se pretende, bastante completa.

A abordagem adotada visa proporcionar o conhecimento dos princípios fundamentais da robótica, seguindo-se uma apresentação das principais aplicações, numa perspetiva de aplicabilidade prática. Desta forma, pretende levar-se os discentes a conhecer e a aumentar a motivação pelos sistemas robóticos com aplicabilidade em sistemas diversos.

5. Demonstração da coerência das metodologias de ensino com os objetivos da unidade curricular

Os objetivos preconizados nesta UC são atingidos mediante a participação nas aulas e o envolvimento em atividades de natureza marcadamente prática, no sentido de proporcionar

uma experiência de aprendizagem completa e uma boa assimilação dos conteúdos. Pretende-se ainda desenvolver as competências dos discentes através da participação nas aulas, e em particular nas aulas teórico-práticas e laboratoriais. Bem como nas visitas de estudo planeadas, e ainda em atividades autónomas com recurso à pesquisa sobre sistemas robóticos.

Os trabalhos desenvolvidos pelos discentes fomentam a pesquisa autónoma bem como o trabalho em equipa, levando a um maior envolvimento e uma maior dedicação facilitando a aprendizagem. A apresentação oral dos trabalhos práticos também contribui para o sucesso na assimilação dos conteúdos e desenvolvimento de competências, não só as competências específicas desta UC, mas também as competências transversais, como medidas facilitadoras de comunicação e expressão de ideias, colaboração interpessoal, entre outras.

6. Conteúdo programático e planificação

O programa da UC de RSI foi desenhado com o intuito do aluno adquirir um conjunto de conhecimentos e competências fundamentais, subjacentes à RSI. A estrutura adotada para as aulas é sequencial, justificando-se, em alguns casos, em virtude de a aquisição de certos conhecimentos e o desenvolvimento de determinadas competências dependerem de outros previamente assimilados. No entanto, no geral, existem conteúdos para os quais a ordem é irrelevante, visto que não requerem conhecimentos e competências anteriores.

O conteúdo programático da UC de RSI está organizado e descrito da seguinte forma:

1. Introdução à robótica inteligente
2. Arquiteturas baseadas em agentes para robôs
3. Perceção e interpretação sensorial
4. Controlo de robôs móveis: locomoção e acção
5. Navegação e localização
6. Localização e mapeamento simultâneos
7. Robótica cooperativa

Na tabela 1 apresenta-se a planificação dos conteúdos programáticos para as aulas de ensino teórico a lecionar durante o semestre.

Em paralelo com as aulas teóricas (T) decorrem as aulas de ensino teórico-prático (TP). Neste tipo de aulas resolvem-se exercícios em que os discentes terão os conhecimentos das aulas teóricas. Para além dos problemas propostos, as aulas de ensino prático (P) fornecem apoio ao desenvolvimento dos trabalhos propostos a realizar no laboratório, os quais constituem as componentes do processo de avaliação periódica dos alunos.

Tabela 1 - Cronograma da distribuição dos conteúdos programáticos de RSI para as aulas de ensino teóricas no semestre letivo.

Aulas	Conteúdos Programáticos
1 - 5	• <i>Introdução à Robótica Inteligente e à Inteligência Artificial</i> • <i>Conceitos Básicos de Robótica</i> • <i>Enquadramento da Inteligência Artificial na Robótica, História, evolução e direções atuais da Robótica Inteligente</i>
5 – 10	• <i>Arquiteturas baseadas em Agentes para Robôs, Reativas, Deliberativas, Híbridas e Cooperativas</i>
10 – 15	• <i>Perceção Robótica, sensores comuns em robótica incluindo visão artificial e profundidade, e técnicas de fusão sensorial</i>
15 – 20	• <i>Localização e Mapeamento, criação, representação a atualização de estados do mundo.</i> • <i>Localização de Markov e Gaussiana</i> • <i>Localização Grid e Monte-Carlo</i> • <i>Mapeamento: Occupancy Grids e SLAM</i>
20 – 25	• <i>Algoritmos de navegação em ambientes conhecidos/Desconhecidos</i> • <i>Diagramas de Voronoi</i> • <i>Algoritmos A* e D*</i> • <i>Decomposição celular</i>
25 – 30	• <i>Introdução à cooperação entre robôs para a realização de tarefas em equipa</i> • <i>Algoritmos PSO/DPSO</i> • <i>Swarm de Robôs</i>

7. Competências a desenvolver

Para os alunos alcançarem os propósitos enunciados na secção 6, estabeleceram-se objetivos específicos para que os alunos adquirissem competências fundamentais. Mais concretamente, constitui objetivo desta UC, os alunos obterem as seguintes competências específicas:

- Compreender os princípios subjacentes da cinemática e da dinâmica;
- Identificar e utilizar as arquiteturas clássicas para controlo em robótica;
- Conhecer o estado da arte em robótica e sistemas inteligentes;
- Definir Autonomia em sistemas robóticos;
- Avaliar a utilização de sistemas de visão comparativamente a outros métodos;

- Aplicar métodos de fusão sensorial, métodos do âmbito da inteligência artificial e métodos de processamento de sinal e de visão para construir perceções do mundo;
- Aplicar métodos de localização, planeamento e navegação em robótica;
- Otimizar as condições de utilização dos sistemas robóticos;
- Solucionar problemas de funcionamento numa instalação fabril;
- Projetar equipamentos para implementação de melhoramentos numa instalação fabril.

Porém, e para além das competências específicas, também durante a frequência desta UC, o discente terá oportunidade de desenvolver e/ou aperfeiçoar outro tipo de competências, designadas competências transversais:

- Capacidade para trabalhar em equipa, de forma coesa e complementar;
- Autonomia na pesquisa bibliográfica sobre um tema proposto;
- Capacidade de analisar, organizar, seleccionar e sintetizar informação recolhida;
- Aptidão para idealizar, estruturar e redigir um bom trabalho;
- Competência para conceber uma boa apresentação do trabalho proposto, tornando-a interessante, objetiva e focando os aspetos essenciais;
- Clareza e fluência na exposição oral.

Para além das competências decorrentes do âmbito dos objetivos, a UC contribui, também, para formar indivíduos que sejam capazes de efetuar trabalho colaborativo e de comunicar com clareza e rigor, sejam autónomos na aprendizagem e que, fundamentalmente, desenvolvam o seu pensamento crítico e capacidade de investigação.

8. Metodologia de ensino

As aulas da UC de RSI estão estruturadas em três tipologias, ensino T, ensino TP e ensino P. A metodologia das aulas de ensino T é, predominantemente, de carácter expositivo, nas quais são apresentados os conceitos referidos na secção 6. Os conceitos, teorias e métodos são expostos e discutidos de modo crítico. Adicionalmente, apresentam-se exemplos ilustrativos dos conteúdos lecionados. O material de apoio à sua leção são essencialmente os diapositivos e o quadro branco.

Por outro lado, nas aulas de ensino TP são resolvidos problemas e analisados casos de estudo. Aqui, espera-se que os alunos tenham um papel ativo na discussão e resolução dos problemas

propostos. Os trabalhos propostos são realizados em grupo e fora do período das aulas. No entanto, as aulas de ensino TP dedicar-se-ão para apoiar o progresso dos trabalhos. Em cada trabalho proposto, os alunos devem incluir uma introdução teórica relativamente às técnicas abordadas.

Espera-se que, com esta metodologia, os alunos desenvolvam competências no projeto e na implementação de RSI, trabalho autónomo, trabalho em grupo e consolidação dos conhecimentos adquiridos durante as aulas de ensino T e TP.

Pretende-se proporcionar a aquisição de conhecimentos e competências por parte dos discentes, através da participação em aulas de natureza teórica (expositivas), teórico-prática (com resolução de problemas) e prática laboratorial (com desenvolvimento de trabalhos de índole marcadamente prática), bem como através da realização de atividades que incluem o recurso à pesquisa sobre sistemas robóticos.

A elaboração de trabalhos de grupo fomenta a pesquisa autónoma bem como o trabalho em equipa, tendo como consequência um maior envolvimento dos discentes e uma maior dedicação, facilitando consequentemente a aprendizagem.

A elaboração de um trabalho de natureza escrita e a apresentação oral do trabalho também contribuem para o sucesso na assimilação dos conteúdos e consequente desenvolvimento das competências pretendidas.

Nas aulas é utilizado o método expositivo recorrendo à projeção das apresentações, e no trabalho laboratorial potencia-se o contacto com as situações que simulam os processos ao nível industrial. São ainda efetuadas visitas de estudo a empresas onde existam tecnologias não possíveis de implementar ao nível laboratorial. Para reforçar a componente aplicada está prevista a realização de um Seminário Industrial.

No contacto com os discentes é privilegiada a utilização de *e-learning* através da plataforma *Moodle*.

9. Metodologia de avaliação

O sistema de avaliação da UC de RSI decorre de acordo com o Regulamento Pedagógico da UTAD [1] podendo ser composto por duas componentes.

Uma componente T é obrigatória, uma componente TP não obrigatória, na qual se avalia os conhecimentos relativos à RSI e uma componente prática P obrigatória, na qual os alunos podem realizar vários trabalhos, a serem desenvolvidos nas aulas, usando as técnicas aprendidas no ensino T, de forma a elaborar um relatório, onde conste uma breve revisão sobre as técnicas aplicadas. Não obstante a realização dos trabalhos seja feita em grupo, as apresentações serão individuais e, conseqüentemente, as notas diferenciadas. Todos os trabalhos terão o mesmo peso para o cálculo da nota final.

A avaliação de conhecimentos far-se-á por exame final com um peso de 40% da nota final (8 em 20 valores), sendo que os restantes 60% (12 valores) são relativos à realização dos trabalhos laboratoriais, ensino P.

As duas componentes da avaliação são independentes e exclusivas em termos de conteúdos: os exames (épocas normal, de recurso, especial ou outras) versam apenas sobre os assuntos expostos nas componentes T e TP, e os trabalhos de laboratório são avaliados pelo desempenho na realização dos trabalhos e na elaboração dos respetivos relatórios.

Dada a obrigatoriedade da parte laboratorial da avaliação e da sua importância na aprendizagem e envolvimento do discente na unidade curricular, entende-se que os discentes devam obter um mínimo de 6 valores (em 12 valores) nesta parte da avaliação a fim de serem considerados aptos para apresentar-se a exame final; os discentes que não obtenham o mínimo atrás referido na avaliação da componente laboratorial estarão automaticamente reprovados.

Desta forma, procura-se motivar os discentes desde do início da unidade curricular, com o objetivo de reduzir ao máximo a fração de discentes desmotivados pela mesma, apenas na proximidade das provas escritas. Cada um dos trabalhos laboratoriais é avaliado com uma nota entre 0 a 100%.

A elaboração de um trabalho em grupo sobre um determinado sistema robótico, o qual deverá abordar todas as questões de simulação e de implementação. Este trabalho deverá ser entregue

e também apresentado oralmente perante toda a turma para discussão. A nota global do trabalho (que resulta da avaliação do trabalho escrito e da apresentação oral) deverá contar para a classificação final em 25%.

Uma componente de avaliação individual respeitante à participação do discente nas aulas (teóricas e práticas-laboratoriais), deverá contar em 15% para a classificação final.

A classificação final, NF, da UC é obtida pela fórmula seguinte:

$$NF = 60\%PE + 40\% \sum T_i \quad (1)$$

onde PE é a nota da prova escrita e T_i é a nota do trabalho prático i , com $i = 1, 2, 3, 4$.

As classificações referidas são atribuídas numa escala de 0 a 20 valores. A classificação final dos alunos deve refletir os seguintes critérios:

- 87.5% – 100.0% ([17.5; 20] valores)
 - o Cumpra ou exceda todas as exigências dos trabalhos atribuídos.
 - o O conhecimento relativo aos conteúdos deve ser excelente.
 - o Apresente um nível e qualidade de conhecimento convincente, tendo realizado uma investigação notável, se aplicável.
- 77.5% – 87.4% ([15.5; 17.5[valores)
 - o Cumpra todas as exigências dos trabalhos atribuídos.
 - o No geral tenha conhecimento preciso dos conteúdos da UC, exceto em alguns pormenores.
 - o Pareça informado sobre os conteúdos, tendo realizado uma investigação adequada quando aplicável.
- 62.5% – 77.4% ([12.5; 15.5[valores)
 - o Cumpra as exigências mais importantes dos trabalhos atribuídos.
 - o Tenha conhecimento preciso na maior parte dos conteúdos lecionados, embora com falhas.
 - o Exiba um conhecimento limitado dos conteúdos, tendo realizado uma investigação limitada quando aplicável.

- 47.5% – 62.4% ([9.5 – 12.5[valores)
 - o Cumpra algumas das exigências mais importantes dos trabalhos atribuídos.
 - o Tenha conhecimento preciso na maior parte dos conteúdos da UC, embora com falhas.
 - o Exiba um conhecimento limitado, tendo realizado uma investigação mínima quando aplicável.
- ≤ 47.4% (< 9.5 valores) (Reprovado)
 - o Não consegue resolver os requisitos importantes dos trabalhos atribuído.
 - o O conhecimento dos conteúdos lecionados no geral é pouco preciso e apresenta falta de qualidade.

10. Repertório bibliográfico

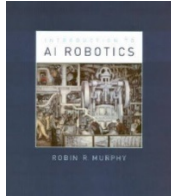
Se por um lado se pretende abordar os assuntos usando exemplos concretos e existindo um total interesse em explorar casos ligados à robótica industrial, por outro lado pretende salvaguardar-se um sólido conhecimento teórico, que permita aos discentes abordar e resolver questões que não tenham explorado nas aulas. A bibliografia abaixo indicada inclui livros de consulta e fonte adicional de apoio para os discentes.

O conjunto bibliográfico disponibilizado permite, facultar aos alunos, a informação que serve de suporte às aulas lecionadas. A bibliografia está dividida em duas partes, bibliografia principal e complementar.

A bibliografia principal serve de suporte às aulas de ensino T ministradas. Apesar da última edição destes livros, serem de 1997 a 2002, eles encontram-se atuais, e abrangem os conteúdos lecionados, os livros são de fácil leitura e compreensão por parte dos discentes. De notar, que estes livros são adotados em muitas unidades curriculares de instituições internacionais. A bibliografia complementar, inclui também obras que abordam os conteúdos programáticos bem como guias e manuais, também em português, para uma melhor aprendizagem do MATLAB® e do ROS que pode ser importante no desenvolvimento dos trabalhos práticos.

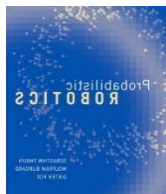
Para além da bibliografia enunciada, é importante referir que os recursos utilizados nas aulas T e TP são disponibilizados no *Moodle* aos alunos, e.g. diapositivos de suporte às aulas, textos de apoio e cadernos de exercícios.

De seguida apresenta-se o repertório bibliográfico principal [Pi] e complementar [Ci] adotado nesta UC:



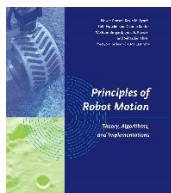
[P1] Murphy, Robin R.; Introduction to AI robotics. ISBN: 0-262-13383-0, 2000.

Este livro agrega a informação fundamental sobre a Inteligência Artificial e é muito claro na apresentação das questões quer ao nível teórico quer ao nível prático, apresentando ainda imensos exemplos reais, minorando a necessidade de produzir textos de apoio para a disciplina.



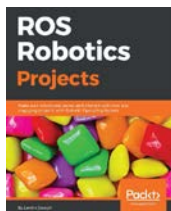
[P2] Thrun, Sebastian; Probabilistic robotics. ISBN: 0-262-20162-3, 2000.

Em cada capítulo deste livro são fornecidos exemplos de implementações em pseudocódigo, e bastante detalhados, sendo relevante para qualquer pessoa envolvida no desenvolvimento de *software* robótico e na investigação científica.



[P3] Choset, Howie; Principles of robot motion. ISBN: 0-262-03327-5, 2007.

Abordagem teórica de fácil leitura que apresenta essencialmente os princípios de movimentação dos sistemas robóticos, demonstrando a capacidade deste tipo de sistemas em ambiente industrial.



[P4] Lentin Joseph; ROS Robotics Projects. ISBN: 978-1-78355-471-3, 2017.

O livro apresenta projetos concretos e motivadores que possibilitam uma aprendizagem mais fácil para os discentes. Abordagem teórica de fácil leitura que apresenta essencialmente a

integração sensorial e da robótica na implementação de sistemas reais, demonstrando a capacidade deste tipo de sistemas em ambiente industrial.

A restante bibliografia está disponível na biblioteca e é complementar e essencial para os discentes que pretendem efetuar o projeto final de curso na área da Robótica.

[C1] Artificial vision for mobile robots: stereo vision and multisensory perception, Nicholas Ayache Cambridge, the MIT, ISBN 0-262-01124-7, 1999.

[C2] Artificial intelligence and mobile robots: case studies of successful robot systems, Ed. by David Kortenkamp, R. Peter Bonasso, and Robin Murphy, Menlo Park, the MIT Press, ISBN 0-262-61137-6, 1998.

[C3] Introduction to robotics: mechanics and control, John J. Craig. - 2nd Ed. - Reading, MA: Addison-Wesley Publishing Company, ISBN 0-201-09528-9, 1995.

[C4] Foundations of robotics: analysis and control, Tsuneo Yoshikawa, Cambridge, MA: The MIT Press, ISBN 0-262-24028-9, 1990.

[C5] Robotics: control, sensing, vision, and intelligence, K. S. Fu, R. C. Gonzalez, C. S. G. Lee., New York, McGraw-Hill Book Company, ISBN 0-07-100421-1, 1987.

[C6] Mobile robotics: a practical introduction, Ulrich Nehmzow, Springer Verlag, ISBN 1-85233-173-9, 1993.

[C7] Mobile robots: inspiration to implementation, Joseph L. Jones, Anita M. Flynn, Bruce A. Seiger, Natick, MA: A K Peters, ISBN 1-56881-097-0, 1999.

[C8] Mobile robots: inspiration to implementation, Joseph L. Jones, Anita M. Flynn, Wellesley, MA: A K Peters, ISBN 1-56881-011-3, 2000.

[C9] Simulação dinâmica e controlo de robôs industriais, Nuno Miguel Fonseca Ferreira, Tese de Mestrado em Engenharia Eletrotécnica e de Computadores, apresentada na Faculdade de Engenharia da Universidade do Porto, 1B-1-8 (ISEC) – 10650, 1999.

11. A unidade curricular em outras instituições de ensino superior

De modo a enquadrar os conteúdos lecionados na UC de RSI, relativamente ao perspectivado em outras instituições de ensino superior, nesta secção comparam-se as UCs de RSI lecionadas em algumas instituições de ensino superior tanto portuguesas como internacionais. Os elementos indicados nesta secção baseiam-se na informação disponibilizada na *world wide web* pelas respetivas instituições de ensino, para o ano letivo 2018/2019.

Tabela 2 - UC de instituições portuguesas e internacionais.

Identificador	Nome da UC	Universidade	Curso licenciatura Mestrado Doutoramento	Ano / Semestre	ECTs	Professor Regente 2018/2019
P1 [2]	Robótica	FEUP	Mestrado Integrado em Engenharia Informática e Computação	5º Ano/1º Semestre	6.0	Armando Jorge Miranda de Sousa
P2 [3]	Sistemas Robóticos e Robótica Móvel	Universidade de Coimbra	Mestrado Integrado em Engenharia Eletrotécnica e de Computadores	4º Ano /1º Semestre	6.0	Jorge Manuel Moreira de Campos Pereira Batista
P3 [4]	Sistemas Autónomos	IST	Mestrado Integrado em Engenharia Eletrotécnica e de Computadores	4º Ano /1º Semestre	6.0	Isabel Ribeiro
P4 [5]	Robótica Móvel	ISEP	Mestrado em Engenharia Eletrotécnica e de Computadores	1º Ano /1º Semestre	3.0	Alfredo Martins
P5	Robótica e Sistemas Inteligentes	UTAD	Mestrado Integrado em Engenharia Eletrotécnica e de Computadores	1º Ano /1º Semestre	6.0	Nuno Ferreira
E1 [6]	Mobile Robots & Navigation	Universitat Politècnica de Catalunya	Master's Degree In Automatic Control And Robotics	1º Ano /1º Semestre	4.5	Juan Andrade Cetto
E2 [7]	Cyber-Physical Systems and Robotics	Universidad Pontificia Comillas ICAI-ICADE	Máster en Industria Conectada	1º Ano /1º Semestre	3.0	Jaime Boal Martín-Larrauri
E3 [8]	Autonomous and Mobile Robotics	Sapienza Università di Roma	Master Programs in "Artificial Intelligence and Robotics" and in "Control Engineering"	1º Ano /1º Semestre	6.0	Giuseppe Oriolo
E4 [9]	Mobile Robotics	University of Essex	Computer Science and Electronic Engineering	5º Ano /1º Semestre	7.5	Huosheng Hu
E5 [10]	Autonomous Mobile Robots	Politecnico de Zurich	Department of Mechanical and Process Engineering	5º Ano /1º ou 2º Semestre	5.0	Roland Siegwart, Margarita Chli, Juan Nieto and Nicholas Lawrance
E6 [11]	Advanced Robotics	University of California at Berkeley	Dept of Electrical Master Engineering & Computer Sciences	5º Ano /1º ou 2º Semestre		Pieter Abbeel
E7 [12]	Robotics: Science and Systems I	MIT	Master of Science Program in Computation for Design and Optimization	5º Ano /1º ou 2º Semestre	12.0	Seth Teller

A tabela 3 pretende comparar o plano curricular desta UC de RSI, proposta neste documento, com UCs similares à de RSI em outras instituições nacionais e estrangeiras, foram analisadas as UCs referenciados na tabela 2.

Tabela 3 - Comparação dos Conteúdos programáticos.

Tópico	P1	P2	P3	P4	P5	E1	E2	E3	E4	E5	E6	E7
<i>Introdução à Robótica Inteligente</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Arquiteturas baseadas em Agentes para Robôs</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
<i>Cinemática e Dinâmica</i>			✓	✓		✓	✓		✓	✓		✓
<i>Perceção e interpretação sensorial</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Localização e Mapeamento</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Controlo de Robôs móveis: Locomoção e Ação</i>	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
<i>Planeamento do Movimento e Trajetórias</i>		✓	✓	✓	✓	✓			✓	✓	✓	✓
<i>Robot Operating System (ROS)</i>					✓	✓	✓					✓
<i>Robôs Humanóides</i>								✓			✓	✓
<i>Navegação e Localização</i>	✓	✓	✓	✓	✓	✓				✓	✓	✓
<i>Localização e Mapeamento Simultâneos</i>	✓	✓	✓		✓	✓				✓	✓	✓
<i>Robótica Cooperativa</i>	✓				✓	✓						✓
<i>Aplicações</i>	✓											✓

Como se pode depreender pela tabela 3, existe uma uniformização nos planos curriculares das diferentes UCs nos diversos cursos, quer nas universidades nacionais quer nas universidades estrangeiras. No entanto, julga-se que a UC proposta neste relatório abrange os tópicos fundamentais de RSI. Além disso, existem temas nos programas de outras instituições que podem ser lecionadas em diferentes unidades curriculares.

As tabelas 4 e 5 apresentam as comparações das metodologias de ensino e das metodologias de avaliação implementadas nas diferentes UCs.

Tabela 4 - Comparação das metodologias de ensino.

Metodologias	P1	P2	P3	P4	P5	E1	E2	E3	E4	E5	E6	E7
Exposição Teórica	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Trabalhos Práticos	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Laboratórios	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓

Tabela 5 - Comparação das metodologias de avaliação.

Metodologias de avaliação	P1	P2	P3	P4	P5	E1	E2	E3	E4	E5	E6	E7
Avaliação distribuída com exame final		50%	30%		40%	40%	40%	50%	60%			
Avaliação distribuída sem exame final	Sim	Não	Não	Sim	Não	Não	Não	Não	Não	Não		
Avaliação continua							10%					15%
Trabalhos Práticos	70%	50%	70%	100%	45%	40%	50%	50%	40%	60%	70%	35%
Trabalho escrito	30%				10%	20%				20%	30%	40%
Defesa Pública dos trabalhos					5%					20%		10%

12. Considerações Finais

Este documento é um relatório sucinto da UC de RSI, apresentando os conteúdos programáticos e as metodologias de ensino e avaliação utilizadas.

Os objetivos da UC assentam na profunda compreensão dos princípios fundamentais da RSI. Assim, os conteúdos programáticos concebidos para a UC visam uma abrangência extremamente ampla desde sistemas simples a sistemas complexos. Os conteúdos estão organizados por partes, de acordo com a tipologia dos assuntos a abordar em causa, e no seu todo constituem uma perspectiva bastante abrangente, nomeadamente, desde as teorias mais clássicas às teorias mais recentes.

É de referir que os conteúdos programáticos desta UC em outras instituições de ensino superior, é muito semelhante, mostrando que esta UC está bem enquadrada e estruturada sendo uma UC de base dos cursos de Engenharia Eletrotécnica entre outros.

As metodologias adotadas visam proporcionar um conhecimento dos princípios fundamentais da RSI e dotar os alunos com as competências específicas e transversais, numa perspectiva de aplicabilidade prática.

Pretende-se ainda desenvolver as competências dos discentes através da participação nas aulas, e em particular nas aulas de ensino T e TP e ainda em atividades autónomas com recurso à pesquisa no ensino P.

Os trabalhos desenvolvidos pelos discentes fomentam a pesquisa autónoma bem como o trabalho em equipa, levando a um maior envolvimento e uma maior dedicação facilitando a aprendizagem. A apresentação oral dos trabalhos práticos também contribui para o sucesso na assimilação dos conteúdos e desenvolvimento de competências, não só as competências específicas desta UC, mas também as competências transversais, como medidas facilitadoras de comunicação e expressão de ideias, colaboração interpessoal, entre outras.

Referências

- [1] Universidade de Trás-os-Montes e Alto Douro: Aviso n.º 9893/2015, de 31 de agosto. Dispõe sobre o regulamento, estrutura curricular e plano de estudos referentes à criação do Mestrado Integrado em Engenharia Eletrotécnica e de Computadores na Universidade de Trás-os-Montes e Alto Douro. Diário da República, 2.ª série (169) (31 agosto 2015) 25127–25137
- [2] Faculdade de Engenharia da Universidade do Porto, unidade curricular de **Robótica**, Mestrado Integrado em Engenharia Informática e Computação, https://sigarra.up.pt/feup/pt/ucurr_geral.ficha_uc_view?pv_ocorrencia_id=399943, [Acedida em 1 março de 2018].
- [3] Universidade de Coimbra, unidade curricular de **Sistemas Robóticos e Robótica Móvel**, Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, <https://apps.uc.pt/courses/PT/unit/8475/14243/2017-2018?type=ram&id=359>, [Acedida em 1 março de 2018].
- [4] Instituto Superior Técnico, unidade curricular de **Sistemas Autónomos**, Mestrado Integrado em Engenharia Eletrotécnica e de Computadores, <https://fenix.tecnico.ulisboa.pt/cursos/meec/curriculo>, [Acedida em 1 março de 2018].
- [5] Instituto Superior de Engenharia do Porto, Instituto Politécnico do Porto, **Robótica Móvel**, Mestrado em Engenharia Eletrotécnica e de Computadores, <https://www.ipp.pt/ensino/cursos/mestrado/isep/105> [Acedida em 1 março de 2018].
- [6] Universitat Politècnica de Catalunya, **Mobile Robots & Navigation**, Master's Degree In Automatic Control And Robotics, <https://www.upc.edu/content/master/guiadocent/pdf/ing/240AR023>, [Acedida em 1 março de 2018].
- [7] Universidad Pontificia Comillas ICAI-ICADE, Cyber-Physical Systems and Robotics, Máster en Industria Conectada, <https://www.comillas.edu/en/cic-en>, [Acedida em 1 março de 2018].
- [8] Sapienza Università di Roma, **Autonomous and Mobile Robotics**, Master Programs in "Artificial Intelligence and Robotics" and in "Control Engineering", <https://www.dis.uniroma1.it/~oriolo/amr/>, [Acedida em 1 março de 2018].
- [9] University of Essex, **Mobile Robotics**, Computer Science and Electronic Engineering, <https://www1.essex.ac.uk/modules/Default.aspx?coursecode=CE315&level=6&period=SP&campus=CO&year=17>, [Acedida em 1 março de 2018].
- [10] Politécnico de Zurich, **Autonomous Mobile Robots**, Master of Mechanical and Process Engineering, http://www.asl.ethz.ch/education/lectures/autonomous_mobile_robots.html, [Acedida em 1 março de 2018].
- [11] University of California at Berkeley, **Advanced Robotics**, Master Engineering & Computer Sciences, <https://people.eecs.berkeley.edu/~pabbeel/cs287-fa15/> [Acedida em 1 março de 2018].
- [12] MIT, **Robotics: Science and Systems I**, Master of Science Program in Computation for Design and Optimization, <https://computationalengineering.mit.edu/cdocurriculum>, [Acedida em 1 março de 2018].

Anexo A – Trabalhos práticos propostos

Os trabalhos práticos da Unidade Curricular de Robótica e Sistemas Inteligentes têm como principal objetivo o desenvolvimento e implementação dos algoritmos que correspondem aos módulos de conteúdos programáticos da UC. A formação dos grupos práticos é definida na primeira aula prática da UC.

Para o ambiente de execução/instalação do sistema será usado o sistema operativo Linux e o sistema operativo ROS. Será disponibilizado um guião de instalação e utilização bem como os detalhes sobre a utilização dos robôs disponíveis no laboratório.

A avaliação dos trabalhos práticos seguirá os seguintes critérios: avaliação individual, inovação, contribuição para trabalho global, e os relatórios. Os relatórios devem ser entregues com antecedência para serem marcadas as defesas nas aulas práticas, incluindo toda a documentação de suporte e código desenvolvido por cada grupo. Deverá ser feita apenas uma submissão por grupo de trabalho.

Neste anexo apresentam-se exemplos dos trabalhos práticos a executar pelos alunos durante o semestre letivo.

Curso: Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Unidade Curricular: Robótica e Sistemas Inteligentes

Ano/Semestre: __ / __

Data: __ / __ / ____

Trabalho nº1

Odometria e equações de movimento de um robô

1. Considere um robot móvel disponível no laboratório. Este desloca-se num plano horizontal por intermédio de quatro rodas motrizes diferenciais.
 - a. Movimente o robô em linha reta de modo a determinar a relação entre os impulsos gerados pelos sensores nas rodas e o deslocamento linear de cada roda.
 - b. Movimente agora o robô apenas com velocidade angular e determine o valor da distância entre as rodas.
 - c. Verifique que pode efetuar movimentos compostos, ou seja em que simultaneamente possuem translação e rotação e com uma só experiência determinar os parâmetros das alíneas anteriores.
 - d. Implemente no simulador em código as equações discretas que estimam a pose do robô (coordenadas x , y , θ), no instante de tempo t_k a partir da pose no ciclo anterior t_{k-1} e dos impulsos recebidos dos encoders associados a cada roda do robô.
 - e. Inicializando a pose estimada do robô na pose inicial do mesmo no simulador, compare a evolução da estimativa da pose do robô dada por estas equações com a verdadeira pose do robô visualizada no simulador. Repare como o erro acumulado aumenta principalmente quando se fazem acelerações e travagens bruscas e as rodas derrapam.

Curso: Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Unidade Curricular: Robótica e Sistemas Inteligentes

Ano/Semestre: __ / __

Data: __ / __ / ____

Trabalho nº2

Implementação de rotinas básicas de controlo do movimento de um robô

1. Ainda considerando o robô do trabalho anterior, suponha que este pode ser comandado indicando a velocidade linear e angular pretendidas. Implemente em código no simulador o procedimento de controlo que recebe estas velocidades e dá como resultado a velocidade de cada roda $v_1(t)$ e $v_2(t)$.
2. Dada a estimativa da pose atual do robô (x_r, y_r, θ_r) e a pose final pretendida (x_f, y_f, θ_f) defina um procedimento que permita ao robô colocar-se da pose final com uma determinada tolerância para o erro angular e da distância.
3. Implemente este procedimento no simulador utilizando uma máquina de estados e teste-o adequadamente. O primeiro estado será rodar para o ponto final até estar alinhado, só depois é que deve avançar para o ponto final. Suponha que o ponto final pode ser alterado em qualquer instante da trajetória.
4. Repita o exercício anterior considerando agora que as variações de velocidades angulares e lineares não são instantâneas, mas considere que existem taxas máximas de aceleração/desaceleração angular e linear.
5. Dado um segmento de reta definido pelos pontos (x_1, y_1) e (x_2, y_2) , defina um procedimento que permita ao robô seguir essa reta no sentido do ponto 1 para o ponto 2, parando neste ponto.
6. Colocando o robô próximo da linha, ajuste um controlador que atue na velocidade de rotação proporcionalmente à distância do robot em relação à linha e ao erro em relação à orientação da linha.
7. Escreva uma função que calcule essa distância. Caso o robô esteja a uma distância elevada, limite a velocidade de rotação do robô. Se estiver a uma distância muito elevada faça primeiro uma trajetória em direção ao ponto mais próximo da linha.

Curso: Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Unidade Curricular: Robótica e Sistemas Inteligentes

Ano/Semestre: __ / __

Data: __ / __ / ____

Trabalho nº3

Localização de robôs móveis baseada em marcos

1. Partindo da situação ilustrada em seguida quer em ambiente simulado quer em ambiente real, onde temos um robô de tração diferencial e limitações detetadas com sensores de visão.
2. Programe uma trajetória em quadrado de modo a que o robô passe junto às paredes e implemente um algoritmo de localização com base nas medias de distância do sensor.
 - a. Adicione ruído às medidas dos sensores e verifique o comportamento do robô e respetiva estimativa da localização. Anote o ruído máximo permitido de modo a que o robô mantenha uma boa estimativa da localização, ou seja, continue a conseguir executar a sua tarefa (percorrer o quadrado).
3. Implemente um Filtro de Kalman Estendido para solucionar o problema de localização com ruído elevado.
4. Utilizando transformações homogêneas, generalize o problema para “marcos” colocados em qualquer posição.



Figura 1 - Setup do laboratório e o Robô.

Curso: Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Unidade Curricular: Robótica e Sistemas Inteligentes

Ano/Semestre: __ / __

Data: __ / __ / ____

Trabalho nº4

Coordenação de múltiplos robôs móveis

1. Considere o sistema apresentado na figura 2, bem como os robôs dos trabalhos anteriores, construa um ambiente real idêntico ao ambiente simulado apresentado na figura 2.
2. Construa o mapa, com a informação de um só robô.
3. Efetuar o mesmo procedimento mas com múltiplos robôs (2 a 3 robôs).
4. Teste diferentes algoritmos e compare os resultados, explique qual a estratégia que adotou para a construção do mapa ambiente, repita o procedimento quer para um quer para múltiplos robôs.

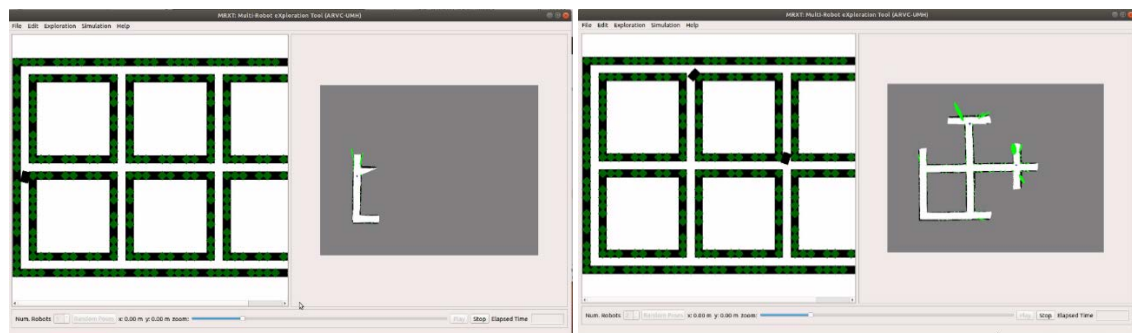


Figura 2 – Exemplo do Setup a construir (construção do mapa por um e dois robôs).

Anexo B – Guião do ROS

De modo a efetuar os trabalhos propostos na cadeira de Robótica e Sistemas Inteligentes, é necessário instalar o ROS no computador.

Este guião exemplifica, para o Sistema Operativo Ubuntu 16.04 LTS, o modo de instalação do ROS, a forma como se deve estabelecer a comunicação entre o ROS e o Simulador Turtlesim, a instalação de vídeo stream e do Arduino, assim como a instalação do sistema de múltiplos robôs do ROS e ainda exemplifica alguns exemplos práticos de aplicação.

B1. Instalação do Sistema Operativo ROS

Para o sistema operativo *Ubuntu* 16.04 LTS, a versão a escolher do ROS é a *Kinetic*. A distribuição correspondente ao sistema operativo *Ubuntu* 16.04 LTS encontra-se no site oficial do sistema operativo ROS (<http://ros.org/install>) apresentado da figura 1.1 e a versão do ROS adotada é o *Kinetic* conforme se apresenta na figura 1.2.

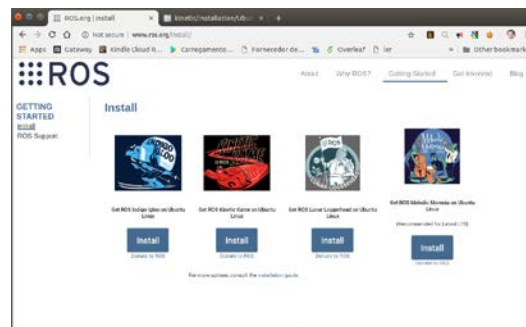


Figura 1.1 - Site oficial do ROS.

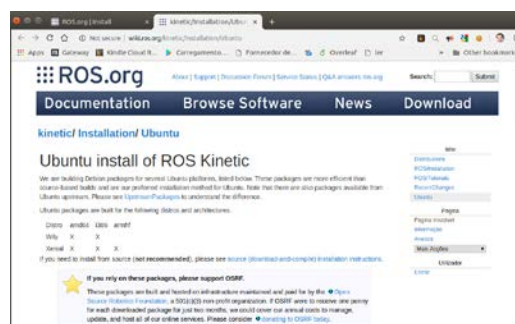


Figura 1.2 - Página da versão *kinetic*.

Para iniciar o processo da instalação do ROS, coloque na linha de comando do *Ubuntu* a seguinte instrução:

Após a instalação é necessário fazer determinadas configurações, bem como ter em atenção os ficheiros que começam por um “ponto” de acordo com a figura 1.3. A razão desta chamada de atenção serve apenas, para informar que é necessário editar o ficheiro `bashrc`.

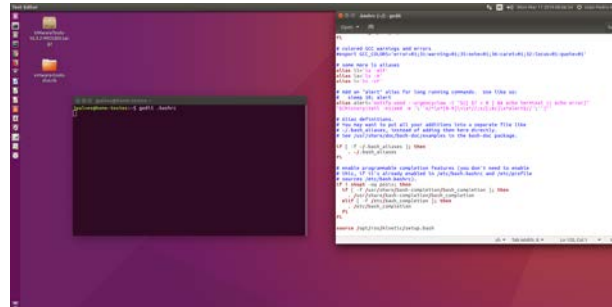


Figura 1.3 - Ficheiro `.bashrc`.

O editor de texto `gedit` faz parte do `gnome` do sistema Linux, no entanto, existem outros editores de texto. Após terminar a instalação, deve existir um ficheiro idêntico ao da figura 1.4 na diretoria `home`, esse documento está assinalado por um `~`.

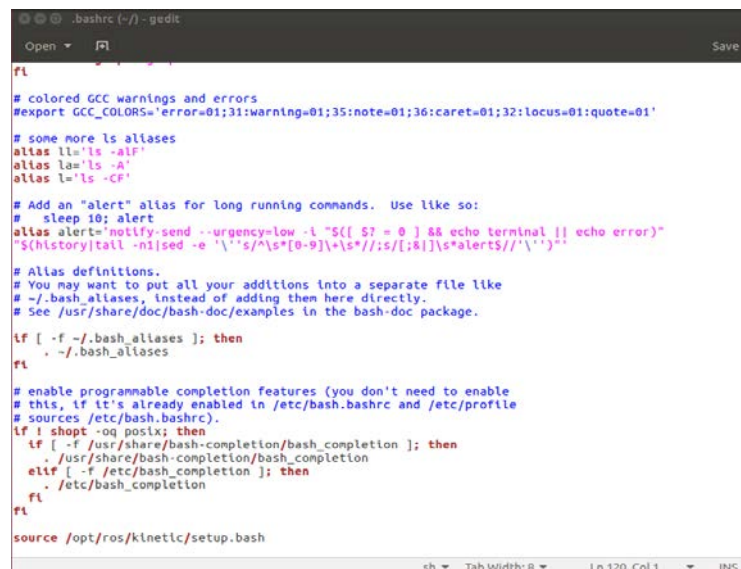


Figura 1.4 - Ficheiro `.bashrc`.

Para configurar o `Workspace` do ROS, é necessário criar uma diretoria chamada `catkin_ws`, sendo necessário utilizar o comando do Linux `mkdir`, este nome `catkin_ws` é atribuído por defeito, mas não é obrigatório mantê-lo. Uma vez criada esta diretoria, é possível entrar com o comando `cd` e de seguida criar uma nova diretoria designada `src` para colocar o código fonte. Após esta configuração, é necessário executar o comando `catkin_init_workspace` de ROS:

catkin_init_workspace

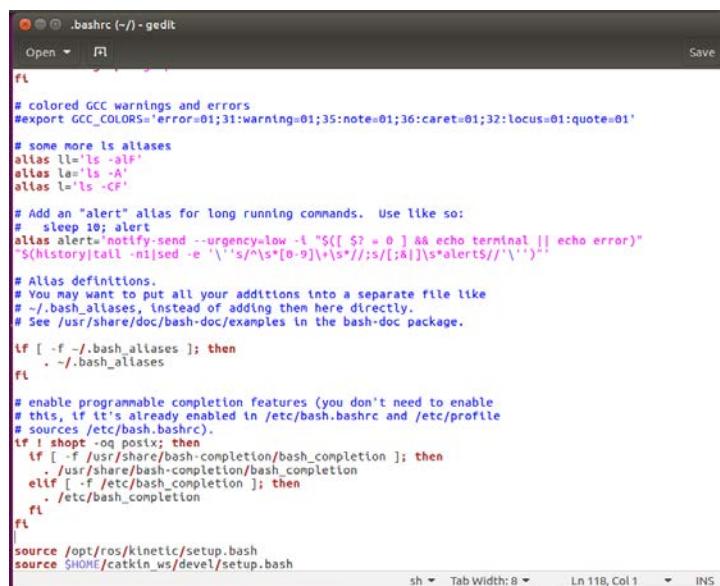
Regressa-se à diretoria do *workspace* do ROS e executa-se o comando:

catkin_make

Este comando *cria* o *workspace*, com todas as modificações finais existentes no ficheiro *.bashrc*, de modo a que o ROS fique operacional e reconhecido pelo MATLAB, para se poderem desenvolver todos os algoritmos necessários para efetuar os trabalhos. Após compilação, acrescente-se uma linha no ficheiro *.bashrc*:

source \$HOME/catkin_ws/devel/setup.bash

Este comando pode ser feito de forma manual ou através do comando *echo*, ficando o *workspace* fica conhecido na linha de comandos.



```
#!/bin/bash
# colored GCC warnings and errors
#export GCC_COLORS='error=01;31:warning=01;35:note=01;36:caret=01;32:locus=01:quote=01'

# some more ls aliases
alias ll='ls -l'
alias la='ls -A'
alias l='ls -CF'

# Add an "alert" alias for long running commands. Use like so:
# sleep 10; alert
alias alert='notify-send --urgency=low -i "${@} $?" && echo terminal || echo error'
"$([history|tail -n1|sed -e 's/^s/[0-9]\s*//;s/[]\s*alert$/\s*alert/' ])"

# Alias definitions.
# You may want to put all your additions into a separate file like
# ~/.bash_aliases, instead of adding them here directly.
# See /usr/share/doc/bash-doc/examples in the bash-doc package.

if [ -f ~/.bash_aliases ]; then
    . ~/.bash_aliases
fi

# enable programmable completion features (you don't need to enable
# this, if it's already enabled in /etc/bash.bashrc and /etc/profile
sources /etc/bash.bashrc).
if ! shopt -oq posix; then
    if [ -f /usr/share/bash-completion/bash_completion ]; then
        . /usr/share/bash-completion/bash_completion
    elif [ -f /etc/bash_completion ]; then
        . /etc/bash_completion
    fi
fi

source /opt/ros/kinetic/setup.bash
source $HOME/catkin_ws/devel/setup.bash
```

Figura 1.5 - Configuração do *workspace*.

Para efetuar a instalação é necessário configurar as variáveis de ambiente do ROS no ficheiro *.bashrc* e preparar o sistema *multimaster*.

export ROS_MASTER_URI=http://home-testes.local:11311

export ROS_HOSTNAME=home-testes.local

O nome da máquina local é *home-testes* e o nome de domínio *router*, tal que estas variáveis têm designações diferentes, consoante a rede onde se trabalha. Para terminar a instalação é

necessário incluir o sistema múltiplos robôs do ROS, para que o sistema fique operacional para a robótica cooperativa.

B2. Comunicação entre o ROS e o simulador Turtlesim

Para estabelecer a comunicação com o simulador, o primeiro passo é estabelecer uma ligação a uma rede ROS através do comando *roscore*. Pode-se iniciar e desenvolver alguns algoritmos, bem como, testar no simulador *turtlesim*, digitando na linha de comandos:

```
\$ roscore
```

Além de ser utilizado o ROS e o simulador *turtlesim* é possível estabelecer uma ligação através do Matlab e o simulador *turtlesim* como se pode observar na figura 2.1.



Figura 2.1 – Turtlesim.

Esta ligação com o simulador *turtlesim* pode ser feita pela janela de comandos, após ter sido executado o comando *roscore*.

```
roslaunch turtlesim turtlesim_node
```

O ROS apresenta diversos tópicos dos quais é possível publicar ou subscrever, para sabermos que tópicos existem disponíveis no simulador, utiliza-se o comando *rostopic list*

```
$ rostopic list
```

```
/rosout
```

```
/rosout_agg
```

```
/turtle1/cmd_vel
```

```
/turtle1/color_sensor
```

```
/turtle1/pose
```

Uma mensagem normalizada do ROS compatível com todos os robôs disponíveis no laboratório, é a utilização do comando `cmd_vel`, que é do tipo `std_msgs/Twist`. Este comando envia para os robôs as velocidades dos motores. O desenvolvimento do *software* dos robôs pode ser feito quer em *Python*, quer em C++ ou em MATLAB (figura 2.2), no caso do MATLAB, não são necessárias mais configurações, porque a *toolbox robotics* permite uma ligação transparente ao ROS.

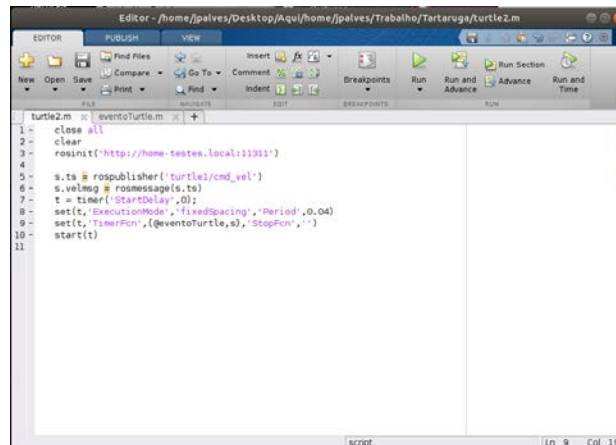


Figura 2.2 - Exemplo de *software* do *turtlesim*.

```
close all
clear
rosinit('http://home-testes.local:11311')
s.ts = rospublisher('turtle1/cmd_vel')
s.velmsg = rosmessage(s.ts)
t = timer('StartDelay',0);
set(t,'ExecutionMode','fixedSpacing','Period',0.04)
set(t,'TimerFcn',{@eventoTurtle,s},'StopFcn','')
start(t)
```

O *software* acima descrito serve apenas para comunicar com o ROS e o Matlab. Por exemplo, o `rosinit`, serve para inicializar a comunicação com um nó do ROS, neste caso o computador é o `home-testes` e o domínio local, a porta 11311, porta do ROS por defeito, podendo ser alterada nas configurações. Ao publicar o `cmd_vel` estamos a enviar uma mensagem de ROS para o simulador e de seguida poderemos enviar mensagens ao `cmd_vel` do simulador.

```
function eventoTurtle(hObject,event,s)
s.velmsg.Linear.X = 0.7;
s.velmsg.Angular.Z = 0.3;
send(s.ts,s.velmsg);
```

Para estabelecer uma comunicação permanente com o ROS cria-se um evento, por exemplo, para alterar a velocidade linear e a velocidade angular. Neste caso, coloque a tartaruga do simulador a fazer um círculo ou outras rotinas mais complexas.

B3. Instalação do vídeo

A instalação do *video stream* pode ser feito através do *github* ou através do site oficial do ROS. http://wiki.ros.org/video_stream_opencv. O comando `git clone <endereço url>`, clona o repositório para o nosso diretório *catkin_ws/src*. Na diretoria *catkin_ws/src* executamos o código fonte: `git clone https://github.com/ros-drivers/video\_stream\_opencv.git`

Após a obtenção de qualquer código fonte é sempre necessário compilar, o comando no ROS a ser usado é o *catkin_make*.

O vídeo poderá ser testado através do MATLAB, e todas as configurações efetuadas são válidas para a raspberry 3B, ou para a raspberry 3B+. O comando *roscore*, inicializa um nó local (master) de ROS.

```
#!/bin/bash
```

```
roscore &
```

```
roslaunch video_stream_opencv camera.launch &
```

```
roslaunch master_discovery_fkie master_discovery_mcast_group:=224.0.0.251 &
```

```
roslaunch master_sync_fkie master_sync
```

O ficheiro apresentado serve para inicializar todos os comandos ao mesmo tempo. O comando *roscore*, onde aparece o símbolo *&*, permite que os próximos comandos sejam também executados. O comando que inicializa o vídeo na *src* do *workspace* do ROS é o *video_stream_opencv* onde existe um *launch* com vários ficheiros, em que um deles se chama *camera.launch*. O comando *roslaunch* permite correr ficheiros do tipo *launch*.

```
roslaunch video_stream_opencv camera.launch &
```

Os restantes comandos destinam-se a sistemas *multi-master*. No MATLAB é necessário subscrever o tópico referente à câmara, por defeito, a mensagem ROS transporta uma imagem a cores de 640x480 mas é possível de ser modificada.

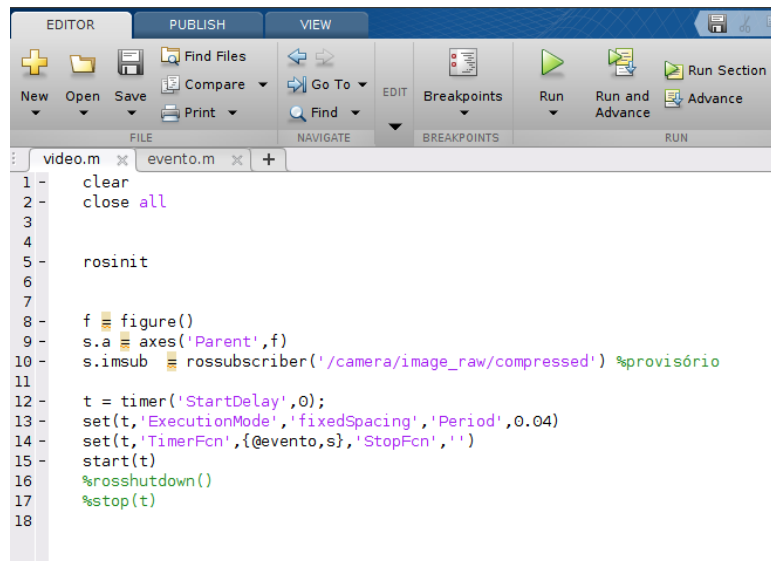


Figura 2.3 – Subscrever um tópico do ROS através do Matlab.

B4. Instalação do Arduino

No site oficial do Arduino, é possível obter o *software* de acordo com o sistema operativo adotado, neste caso Linux de 64 bit, podendo ser verificado com o comando `uname -mir`. Através do comando `install.sh`, efetua-se a instalação e de seguida cria-se uma diretoria Arduino bem como sub-diretorias *libraries*, sendo necessário instalar a *bridge ros arduino* para estabelecer a comunicação com o ROS.

```
sudo apt install ros-kinetic-rosserial-arduino
```

```
sudo apt install ros-kinetic-rosserial
```

Na diretoria `Arduino/libraries` é necessário executar o comando `roslaunch rosserial_arduino make_libraries.py` para criar as bibliotecas que unem os sistemas operativos ROS ao Arduino, sendo necessário obter as permissões para aceder à porta série `as tty`

```
sudo usermod -a -G dialout $USER
```

Após a instalação é necessário reinicializar (*reboot*) o computador ou a *raspberry*. Antes de iniciar o desenvolvimento dos algoritmos é necessário estabelecer uma ligação via ssh.

```
ssh utilizador@nome_da_máquina.dominio
```

A utilização do ssh é devido à reduzida largura de banda e é essencial em redes de pequena escala com recursos limitados, em especial para ser possível trabalhar com um sistema *multi-master*, isto porque, circulam diversas mensagens de vídeo que ocupam grande parte da largura de banda. Do mesmo modo, para controlar o robô podemos enviar comandos do tipo *cmd_vel* mas é necessário criar um *publisher*, para enviar mensagens para o Arduino. Como os vídeos do ROS trabalham com OpenCV e o desenvolvimento será feito em Matlab é necessário converter as imagens para RGB.

B5. Exemplos práticos

Após estas notas introdutórias, é possível estabelecer a ligação a qualquer uma das *raspberry* que se encontram nos diferentes robôs que existem no laboratório e começar a desenvolver os trabalhos propostos.

No exemplo que se pretende apresentar, é possível ler uma mensagem de vídeo, sendo necessário subscrever a câmara do robô (*robot 3*) que está ligada à *raspberry*. Após a inicialização do nó de Matlab através do comando *rosinit* é necessário subscrever este tópico:

```
rosinit
f = figure()
s.a = axes('Parent',f)
s.imsub = rossubscriber('/robot3/image_raw/compressed');
```

Em seguida, pode-se publicar o *cmd_vel* e chamar este evento, por exemplo, de 40 em 40 ms, o qual comunicará com o robô estabelecendo a comunicação da imagem bem como das mensagens de comandos de velocidade que seguem para o robô 3.

```
s.ts2 = rospublisher('/robot3/cmd_vel')
s.velmsg2 = rosmesssage(s.ts2)
t = timer('StartDelay',0);
set(t,'ExecutionMode','fixedSpacing','Period',0.04)
set(t,'TimerFcn',{@evento1,s},'StopFcn','')
```

Para demonstrar as funcionalidades descritas vamos mostrar um algoritmo para reconhecer uma bola vermelha existente no laboratório, e porque queremos manter a simplicidade no algoritmo que pretendemos desenvolver, vamos utilizar a transformada circular de Hough e um

filtro básico para vermelhos sem ser necessário segmentar as cores da imagem. No entanto, é necessário subscrever diversos tópicos do robô, a função *receive* recebe a mensagem ROS do subsciber de vídeo, e depois transforma-se num tensor RGB através da função *readImage* para detetar a bola vermelha no Matlab, e o filtro de vermelhos é conseguido com base de uma simples subtração da matriz dos vermelhos pela matriz dos azuis, e compensamos o efeito do brilho da imagem iremos somar uma função adaptativa com base na média e na função sigmóide. Em seguida, calcula-se a transformada de Hough e através da função *imfindcircles* calcula-se o centróide do raio da circunferência e obtém-se o ponto médio da bola, com estas informações já poderemos enviar ordens para movimentar o robô, por exemplo, para seguir a bola ou para fugir dela.

B6. Instalação do sistema multi-robôs no ROS

O sistema *multi-master* não vem instalado por defeito no ROS, para o instalar coloca-se na linha de comandos a seguinte instrução:

```
sudo apt install ros-kinetic-multimaster-fkie
```

Com o sistema *multi-master* podemos unir várias redes ROS numa só, os tópicos são transmitidos e sincronizados via *multi-master*, isto significa que podemos estar ligados ao nosso *localhost* e mesmo assim receber tópicos de máquinas distantes.

```
roslaunch master_discovery_fkie master_discovery_mcast_group:=224.0.0.251 &  
roslaunch master_sync_fkie master_sync
```

Para saber qual o IP do *multicast* da rede, utiliza-se o comando *netstat -g*.

Para demonstrar, iremos testar o sistema com duas *turtlesim* no simulador. E depois repetimos o processo mas com os robôs num cenário real. Executando o seguinte comando abrem-se duas janelas com duas *turtlesim*:

```
roslaunch turtlesim multisim.launch
```

Se previamente já tiver executado as instruções de *multi-master*, os tópicos vão ser publicados em toda a rede, e podemos observá-los com o *node_manager*.

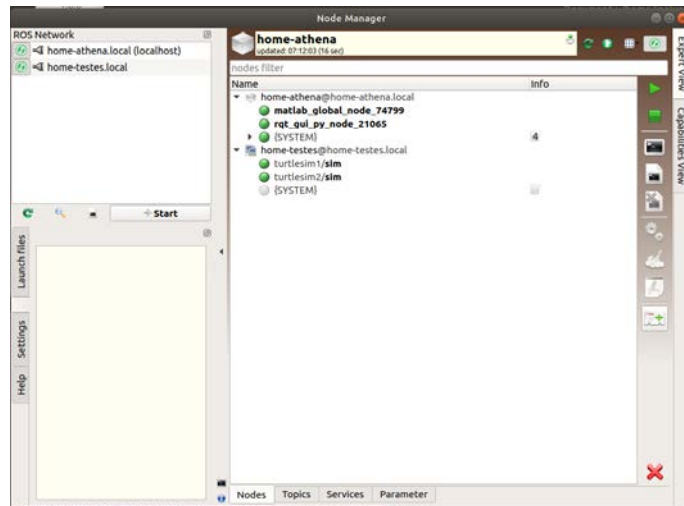


Figura 2.4 - Rede ROS.

O código a desenvolver em Matlab® é relativamente simples: publicamos os dois tópicos do *turtlesim* , no *node_manager* , com dois *namespaces* diferentes, para não haver conflito de comunicação entre os dois robôs. No nosso caso, enviamos mensagens ao *cmd_vel* e assim poderemos controlar um enxame de robots, fazer SLAM, tal que desta forma, todos os robôs comunicam uns com os outros, possibilitando assim um sistema distribuído.

close all

clear

rosinit

```
ts1 = rospublisher('turtlesim1/turtle1/cmd_vel')
s.ts2 = rospublisher('turtlesim2/turtle1/cmd_vel')
s.velmsg1 = rosmesssage(s.ts1)
s.velmsg2 = rosmesssage(s.ts2)
t = timer('StartDelay',0);
set(t,'ExecutionMode','fixedSpacing','Period',0.04)
set(t,'TimerFcn',{@eventoTurtle,s},'StopFcn','')
start(t)
```

No evento seguinte, controlamos separadamente as velocidades (angular e linear).

```
function eventoTurtle1(hObject,event,s)
```

```
s.velmsg1.Linear.X = 0.7;
```

```
s.velmsg1.Angular.Z = 0.3;
```

```
send(s.ts1,s.velmsg1);
```

```
s.velmsg2.Linear.X = -0.7;
```

```
s.velmsg2.Angular.Z = 0.3;
```

```
send(s.ts2,s.velmsg2);
```

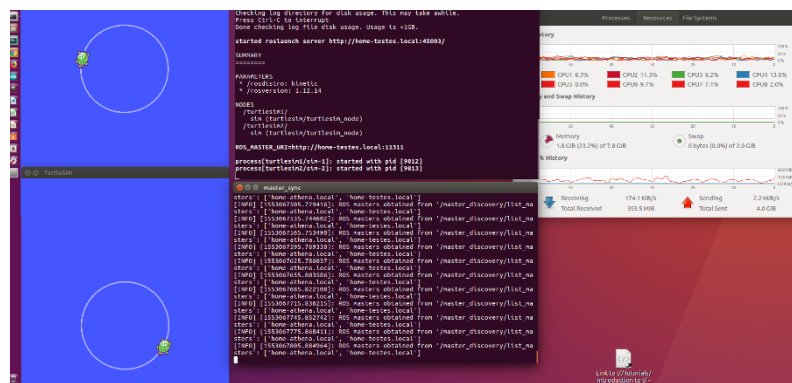


Figura 2.5 – Acesso ao sistema com duas *turtlesim* no simulador.

Anexo C – Exemplo de Exame de Avaliação

Universidade de Trás-os-Montes e Alto Douro Escola de Ciências e Tecnologia Departamento de Engenharias

Exame da Época Normal

Curso: Mestrado Integrado em Engenharia Eletrotécnica e de Computadores

Unidade Curricular: Robótica e Sistemas Inteligentes

Ano/Semestre: __ / __

Data: __ / __ / ____

Nome : _____

Aluno Nº: _____

É obrigatória a apresentação de documento de identificação com fotografia sempre que o docente encarregado da vigilância da prova o solicitar.

O Exame é sem consulta e a duração da prova é de 120 min. Por favor, leia o exame completo antes de preparar as respostas para se certificar que tem uma boa visão das questões a serem respondidas. Boa sorte!

1. Qual é a principal dificuldade em utilizar um acelerómetro para obter a posição relativa? Qual é a causa e resultado deste problema? (5p)
2. Mencione três causas para os erros sistemáticos na odometria. (5p)
3. Explique a diferença entre *tracking* e localização global. (10p)
4. Podem-se utilizar métodos de planeamento global com realimentação dos sensores? Se sim, como? (5p)
5. Qual é a diferença entre os sensores propriocetivos e exterocepetivos? (5p)
6. Mencione três princípios para medir o alcance de objetos? (10p)
7. Qual é a motivação da utilização de sensores nos robôs? (5p)
8. O algoritmo SLAM (*Simultaneous Localization And Mapping*) é uma solução para o problema do ovo e da galinha. Explique brevemente esta afirmação. (10p)
9. O que mede o sensor de ultrassons? Como se processa a medição? Caracterize os prós e os contras do uso deste tipo de sensor. (10p)

10. Na construção de um robô móvel e para ter certeza na escolha de um *encoder* é suficiente que detete se mudou 0,1 mm. As rodas da plataforma têm um raio de 100 mm sem carga. A engrenagem, a caixa entre o motor e a roda é 1:20. A carga da plataforma comprime as rodas em 10%. Qual resolução do *encoder* e onde se coloca? (20p)
11. Pretende-se controlar um robô através de um computador e que seja capaz de seguir uma estrada.
- a. Quais os sensores a utilizar e porquê? (5p)
 - b. Apresente os maiores desafios para que sistema funcione num cenário real. (5p)

Anexo D – Plano curricular do MIEEC - UTAD

	1º Semestre	2º Semestre
1º ano	Álgebra linear e geom. analítica Análise matemática I Fundamentos de eletrotecnia Práticas laboratoriais Programação de computadores Seminário em ciências da eletrotecnia e computação	Análise matemática II Mecânica e ondas Programação por objetos Sistemas digitais Sistemas e sinais Teoria dos circuitos
2º ano	Análise matemática complementar Eletrónica I Probabilidades e estatística Sistemas elétricos de energia Termodinâmica e materiais elétricos	Algoritmia Arquitetura de computadores Eletromagnetismo e ótica Eletrónica II Matemática computacional
3º ano	Comunicação de dados Eletrónica de potência Eletrónica digital Máquina elétricas Sistemas de controlo	Fundamentos de telecomunicações Instalações elétricas Instrumentação e sensores Projeto em Eng^a. Eletrotécnica e de computadores Redes de computadores
4º ano	Automação industrial Gestão de projetos em engenharia eletrotécnica Opção A-1 Opção B-1 Opção C-1 Otimização e algoritmos Processamento digital de sinal Sistemas de microcontroladores	Antenas * Aplicações de eletrónica e instrumentação * Controlo digital Gestão da qualidade da energia * Gestão de redes e sistemas distribuídos * Microsensores e microatuadores * Opção A-2 Opção B-2 Opção C-2 Robótica * Sistemas de informação e bases de dados * Sistemas inteligentes
5º ano	Aplicações e serviços web * Biotelemetria * Comunicações móveis e redes sem fios * Controlo avançado * Energias alternativas * Microeletrónica * Opção A-3 Opção B-3 Opção C-3 Dissertação - mestrado integrado em engenharia eletrotécnica e d (5054)	

Opcionais	Opção A-1	Opção A-2	Opção B-1	Opção B-2	Opção C-1	Opção C-2	Opção A-3	Opção B-3	Opção C-3
Computação evolutiva*						X			
Computação gráfica		X		X		X			
Computação ubíqua						X			
Instalações elétricas avançadas*				X					
Processamento digital de imagem	X		X		X				
Produção e transporte de energia*			X						
Redes avançadas de computadores*					X				
Redes de distribuição de energia*				X					
Segurança em instalações elétricas*			X						
Técnicas de comunicação multimédia*					X				
Tópicos avançados de eletrónica	X								

Legenda:

* - Unidade Curricular pertencente a um ramo

Fonte: <https://www.utad.pt/estudar/cursos/engenharia-eletronica-e-de-computadores/>