

ROSI CHALLENGE

Regulamento do desafio

Navegação autônoma de plataforma robótica móvel para inspeção
de transportadores de correia

Versão 1.0

Sumário

Introdução	5
Descrição do desafio	6
Plataforma robótica - ROSI	7
Ambiente de Inspeção	8
Tarefas de Inspeção, Pontuações e Penalizações	9
Estrutura do desafio	17
Etapas	17
1ª Etapa - Pré-avaliação	17
2ª Etapa - Qualificatória/Eliminatória	18
3ª Etapa - Final	18
Datas do Desafio	19
Classificação Final e Premiação	19
Inscrições	21
Demais Informações do Desafio	23
Comissão organizadora e técnica	23
Comissão de apoio	23
Comitê avaliador	23
Disposições gerais	24
Contato	24

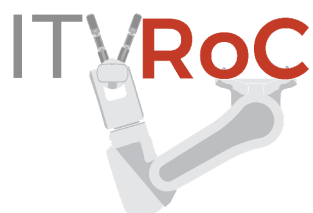
ROSI CHALLENGE

Patrocinadora



Organização e Realização

INSTITUTO TECNOLÓGICO VALE



Laboratório de Controle, Robótica e Instrumentação
Instituto Tecnológico Vale
Ouro Preto



GSCAR

Grupo de Simulação e Controle em
Automação e Robótica
COPPEIUFRJ

Apoio



Introdução

O ROSI (Robô para Serviços de Inspeção) é um dispositivo robótico da VALE para inspecionar Transportadores de Correia. O nome é uma homenagem a Rosimar Cucco, primeira mantenedora a trabalhar no Porto de Tubarão da VALE, em Vitória-ES.

Os Transportadores de Correia são estruturas que podem chegar a quilômetros de extensão; no caso da Vale, estas são responsáveis pelo transporte diário de milhares de toneladas de minério. Realizada majoritariamente por operadores, a inspeção de suas condições é dificultada devido a sua grande extensão e riscos envolvidos na tarefa. Sendo assim, o ROSI tem como objetivo permitir aos operadores gerenciar de forma segura a inspeção remota destas estruturas, além de aumentar a eficiência das análises através da coleta de dados de forma padronizada.

O objetivo do ROSI CHALLENGE é realizar a inspeção autônoma de um Transportador de Correia utilizando uma plataforma móvel com locomoção reconfigurável, braço manipulador de 6 graus de liberdade e conjunto de sensores. Serão propostas tarefas de inspeção, pontuadas de acordo com a complexidade da sua execução.

Todos os códigos desenvolvidos, seja pela organização ou pelas equipes, serão disponibilizados a comunidade ao final do desafio sob a licença GNU-GPL 3.0 em repositórios Github. O simulador em si também é um legado pois proporciona um ambiente poderoso de simulação, podendo ser utilizado inclusive para aulas práticas de robótica. Adicionalmente, as soluções das equipes poderão ser incorporadas no protótipo do ROSI, proporcionando reconhecimento institucional às equipes contribuintes.

Descrição do desafio

O desafio ocorrerá **integralmente** em ambiente simulado utilizando o **V-REP**. O simulador do ROSI, bem como as instruções de instalação e outras informações, se encontram no repositório:

<https://github.com/filRocha/rosiChallenge-sbai2019>

O simulador se comunica com o **Robot Operating System** (ROS) através da interface **vrep_ros_interface**. Os códigos desenvolvidos devem ser estruturados como um **único** pacote do ROS. Caso necessite de mais pacotes, estruture-os sob um pacote principal. O sistema operacional oficial do desafio é o **Ubuntu 18.04.2 LTS (Bionic Beaver)**. Atente-se quanto aos pacotes dos repositórios padrão suportados por esta versão.

As equipes devem subir seu pacote ROS para, em um primeiro momento, um repositório privado do Github sob licença **GNU General Public License v3.0** - repositórios privados podem ser criados gratuitamente utilizando contas de estudante do Github. O intuito de ser privado no primeiro momento é evitar o vazamento das soluções entre as equipes. Visando o compartilhamento das soluções com a comunidade, será obrigatória a mudança do repositório para **público** durante a etapa final do desafio. Será instruída a adição de uma conta oficial da organização como colaboradora do repositório da equipe a fim de viabilizar a coleta dos códigos durante as etapas de avaliação.

Para avaliar o algoritmo durante as etapas, a organização irá coletar o pacote do repositório Github das equipes, compilar o **catkin_ws** utilizando a ferramenta **catkin build** e executar um **único arquivo launch** indicado pelo time. Poderão ser fornecidas curtas e claras instruções de instalação de pacotes/dependências no **README.md** do repositório da equipe. Entretanto, as dependências são restritas a instalações através do **apt** e **pip** considerando apenas os **repositórios padrão** do **Ubuntu 18.04.2**, **ROS Melodic Morenia** e **Python 2.7**, i.e. não poderão ser adicionados repositórios customizados. A equipe deve se atentar quanto à eficácia das instruções, pois a não reprodutibilidade de alguma etapa poderá levar à eliminação do time. Logo, observe sempre a **portabilidade** de seu código.

É permitida (inclusive sugerida) a utilização de pacotes públicos do ROS disponibilizados por terceiros. Entretanto, os mesmos devem estar contidos sob a organização do pacote principal da equipe e utilizando a recursividade do git para a referência durante o processo de clonagem - veja material de referência do ROS e GIT sobre isto. Fique sempre atento quanto à factibilidade de instalação e dependências destes pacotes terceiros.

Como um dos objetivos do desafio consiste no compartilhamento das soluções com a comunidade, a **legibilidade** do código também será avaliada. A desconsideração de boas práticas de programação, como a ausência de comentários em todas as seções do código, poderá levar a penalização da nota final. Adicionalmente, a instalação e uso do código deverá estar bem catalogado no **README.md** do repositório da equipe.

Todas as etapas do desafio poderão ser realizadas **remotamente**, não necessitando da presença *in loco* dos membros da equipe. A apuração da última etapa e divulgação da classificação final ocorrerá durante o SBAI XIV, a ser realizado no Centro de Convenções da UFOP, Ouro Preto-MG, de 27 a 30 de outubro de 2019. A presença das equipes durante a etapa final é desejada, porém **não obrigatória**.

Plataforma robótica - ROSI

O ROSI é uma plataforma móvel robótica composta por um conjunto de sensores para realização das inspeções, um manipulador Universal Robots UR-5 com 6 DoF, além de uma plataforma móvel com sistema de locomoção reconfigurável.

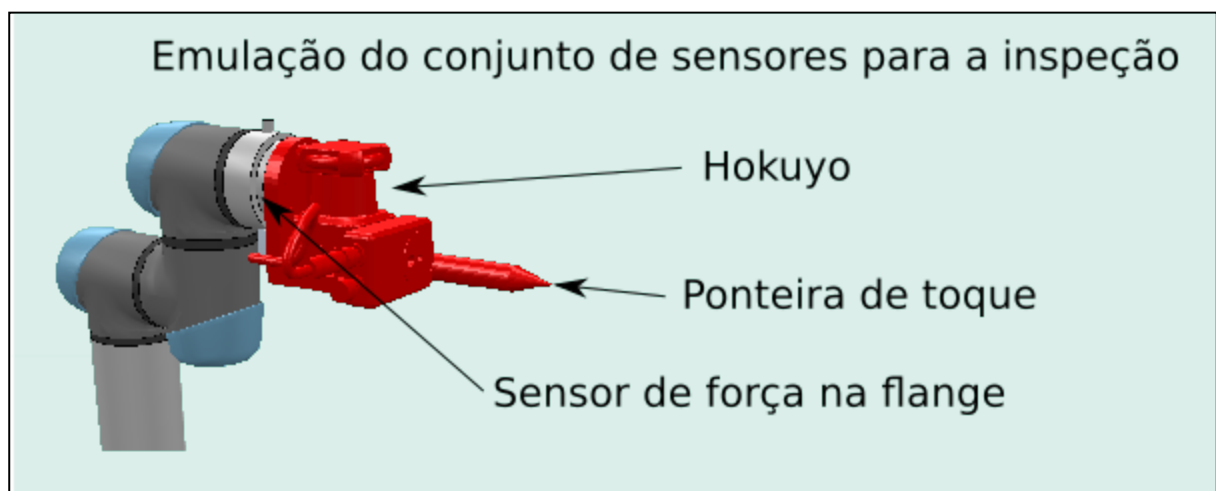
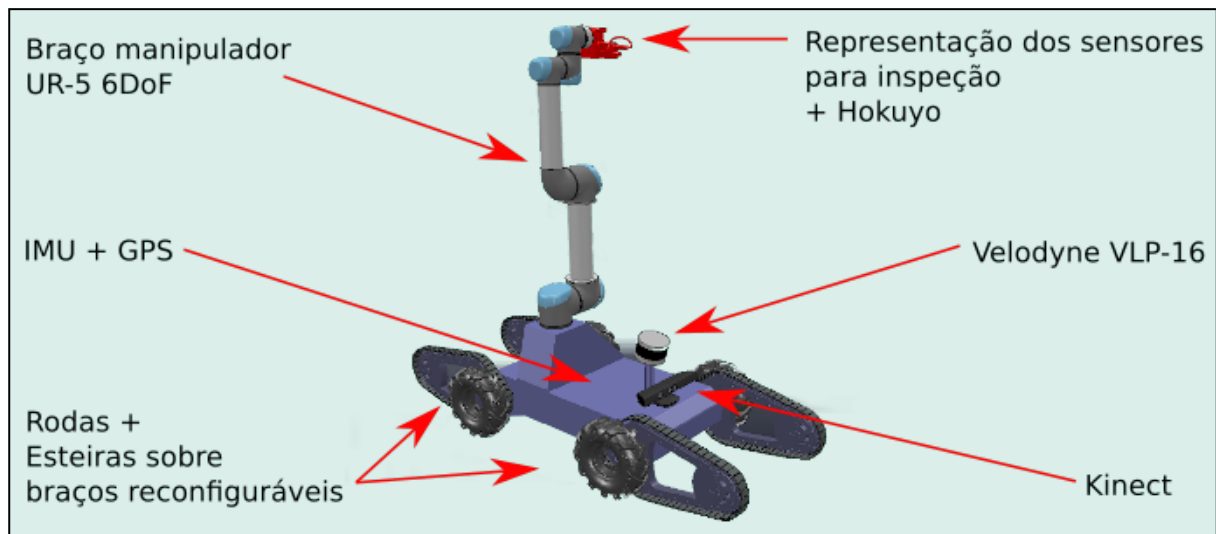
Como sensoramento, o robô conta com:

- **Kinect:** câmera RGB com luz estruturada que fornece imagens com informação de profundidade; possui uma junta ativa horizontal em sua base para orientar seu campo de visão.
- **Velodyne VLP-16:** 16 feixes de laser rotacionais fixados na base do robô;
- **IMU:** unidade de sensoramento inercial;
- **GPS:** fornece uma localização ruidosa da plataforma móvel;
- **Hokuyo:** um feixe de laser rotacional localizado na flange do manipulador robótico;
- **Ponteira de toque:** emulação do sensor de vibração que faz coletas da estrutura do TC.
- **Sensor de força:** localizado na flange do manipulador robótico; mede a força de compressão da ponteira de toque com a estrutura tocada.

Como atuadores, é possível comandar:

- Velocidade rotacional dos braços reconfiguráveis envoltos pelas esteiras de tração.
- Velocidade rotacional das juntas do sistema de tração; aciona simultaneamente as rodas e as esteiras.
- Posição angular da base do Kinect.
- Posição angular das juntas do manipulador UR-5.

Todas as unidades de medida seguem o SI.

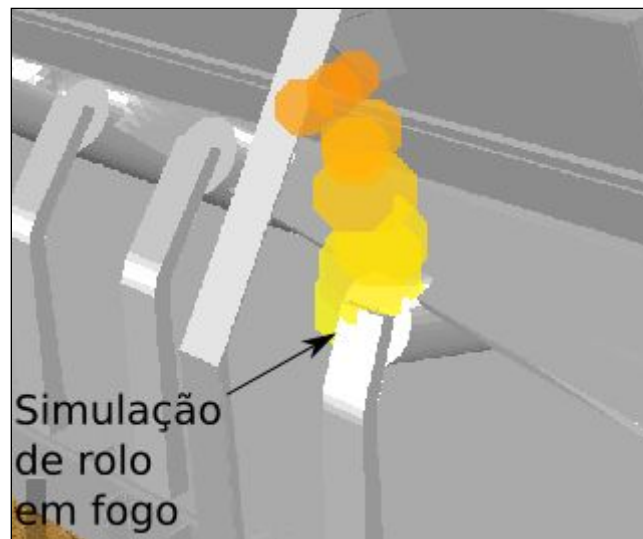
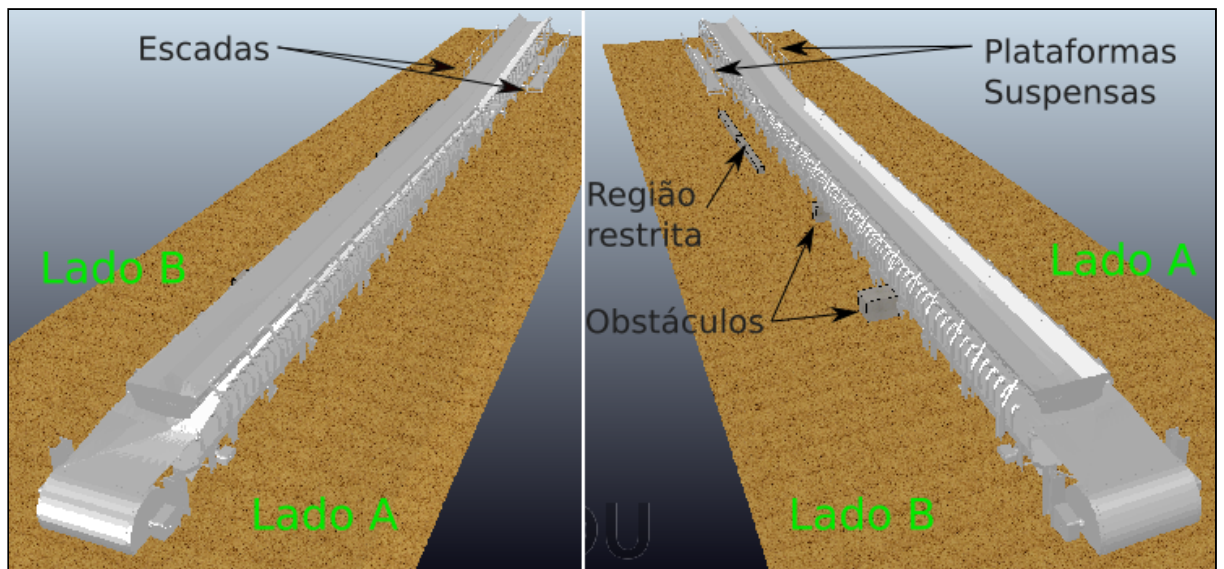


As informações de como interagir com estes elementos no ROS estão descritas no [README.md](#) do repositório do simulador, bem como no código ROS de exemplo [rosi_joy.py](#).

Ambiente de Inspeção

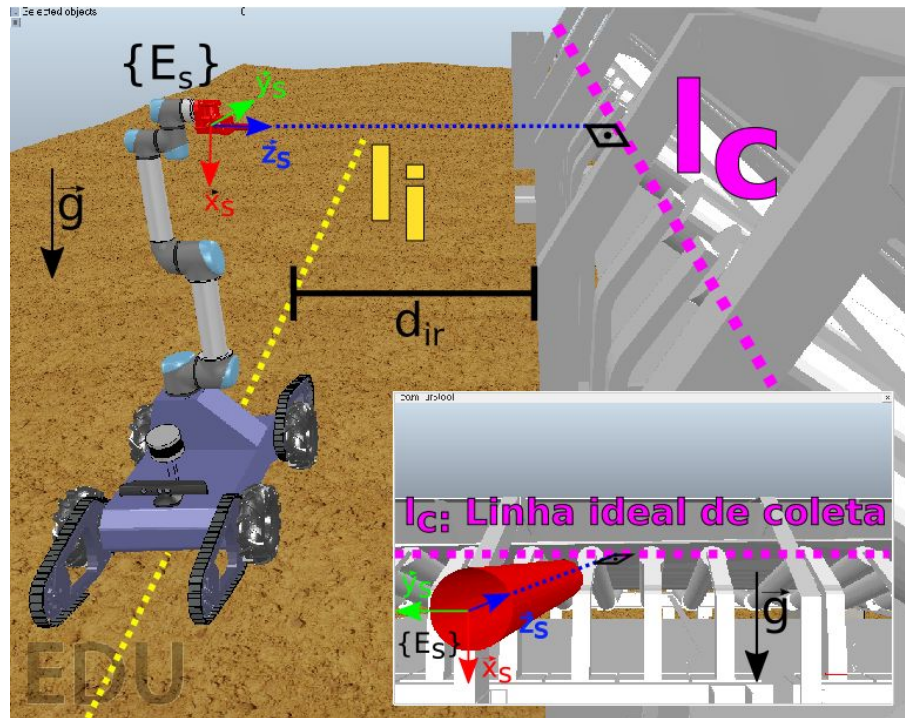
O ambiente de inspeção consiste em um trecho de Transportador de Correia sobre um terreno não estruturado. O TC possui duas regiões com plataformas suspensas para a locomoção dos inspetores em regiões elevadas. O simulador apresenta dois rolos pegando fogo, sendo um em região térrea e outro na plataforma suspensa. Há também uma região restrita no térreo, além de um obstáculo menor e outro maior.

As tarefas específicas a serem realizadas, além de suas pontuações, são descritas na próxima seção.



Tarefas de Inspeção, Pontuações e Penalizações

O ROSI deve realizar tarefas específicas de inspeção em TC de forma autônoma e dinâmica. Por dinâmica, entende-se completar as tarefas designadas independentemente da disposição das mesmas no ambiente. A todo momento é solicitado que o robô realize **coleta** de imagens dos rolos do TC.



A figura acima descreve o conceito de **coleta** considerado nesta competição. Traça-se uma linha imaginária l_c ligando os eixos dos rolos. O caminho percorrido pelo robô durante a inspeção é descrito por uma linha imaginária denominada l_i . Fixa-se um sistema de coordenadas $\{E_s\}$ solidário ao conjunto de sensores de inspeção - a fixação da origem de $\{E_s\}$ aqui é **arbitrária** e serve apenas para **exemplificação**; as equipes devem fixar o sistema de coordenadas da maneira que acharem mais conveniente. Considerando o $\{E_s\}$ aqui fixado, o eixo \vec{z}_s , alinhado à ponteira de toque, deve ser orientado de forma a manter um ângulo de 90° com l_c . O vetor \vec{x}_s deve ser mantido alinhado ao vetor de gravidade \vec{g} . A distância d_{ir} entre l_i e a base do TC não deve ser maior que 4 metros, salvo situações onde a presença de obstáculos impeça que esta restrição seja respeitada.

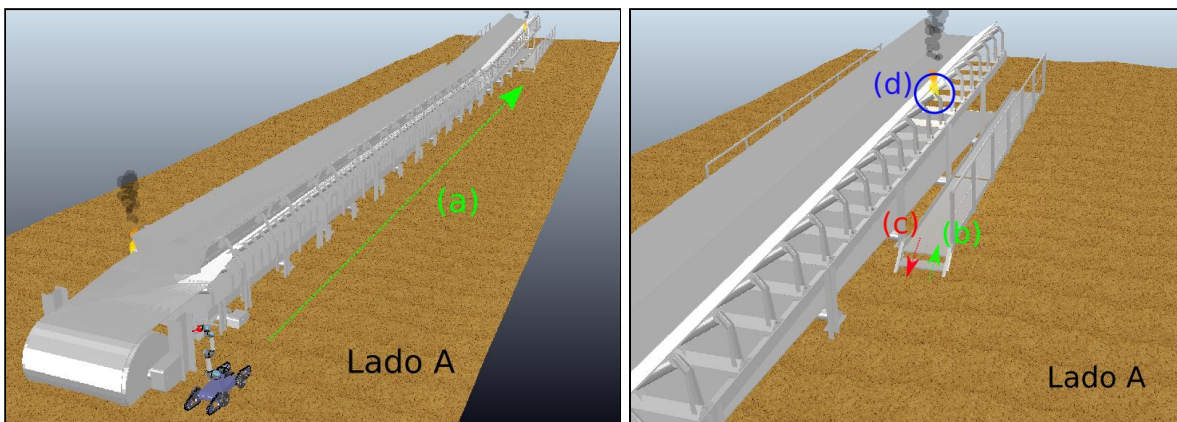
A Tabela 1 lista as tarefas específicas requeridas, bem como suas respectivas **pontuações base**. As tarefas indicadas podem ter sua pontuação multiplicada por um fator $k_c = \{k_c \in \mathbb{R} \mid 1,0 \leq k_c \leq 1,5\}$ em função do quão correta a **coleta** é realizada durante a realização da tarefa. As que não forem passíveis de k_c obterão automaticamente $k_c = 1$ na fórmula de nota. Ademais, um fator $k_b = \{k_b \in \mathbb{R} \mid 0 \leq k_b \leq 1\}$ será multiplicado à pontuação base de todas as tarefas em função do **quão correta** a tarefa em específico foi realizada. As referências de localização para execução das tarefas são ilustradas em imagens posteriores.

Tabela 1 - Tarefas de inspeção

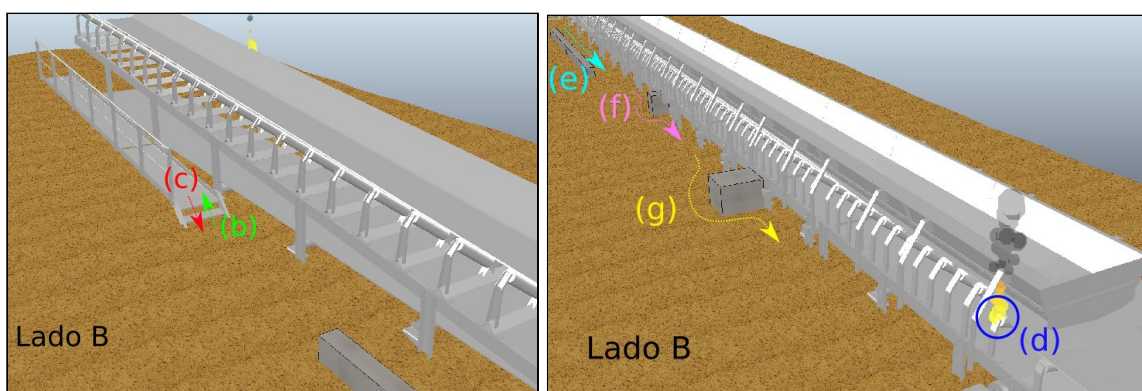
Tarefa	Referência	Descrição	Pontuação
T-01	-	Realizar todas as demais tarefas desta tabela com $k_b \geq 0,3$ em todas.	25
T-02	Rolo em fogo como em (d). Região de cavalete como em (h)	Tocar o cavalete do rolo em fogo sobre a plataforma suspensa: após identificado algum rolo em chamas no trecho da plataforma suspensa, o ROSI deverá tocar a região do cavalete com a ponteira de toque. O toque ideal é feito com o vetor \vec{z}_s^+ alinhado ao vetor normal ao plano do cavalete. A ponteira de toque possui uma complacência elástica; entretanto, a força de compressão no punho não pode passar de 1 N. Esta força será avaliada pelo sensor de força e torque localizado na flange do manipulador robótico.	20
T-03	Trajeto como em (b)	Subir escada: o ROSI deverá subir uma escada de forma controlada, sem tocar qualquer parte da estrutura em seu entorno.	15 passível de k_c
T-04	Trajeto como em (c)	Descer escada: o ROSI deverá descer uma escada de forma controlada, sem tocar qualquer parte da estrutura em seu entorno.	10 passível de k_c
T-05	Rolo em fogo como em (d). Região de cavalete como em (h)	Tocar o cavalete do rolo em fogo no térreo: após identificado o rolo em chamas em região térrea, o ROSI deverá tocar a região do cavalete com a ponteira de toque. O toque ideal é feito com o vetor \vec{z}_s^+ alinhado ao vetor normal ao plano do cavalete. A ponteira de toque possui uma complacência elástica; entretanto, a força de compressão no punho não pode passar de 1 N. Esta força será avaliada pelo sensor de força e torque localizado na flange do manipulador robótico.	10
T-06	Trajeto como em (g)	Coleta contínua desviando do obstáculo maior: enquanto realizando uma coleta retilínea ao lado do TC, o ROSI deverá desviar do obstáculo maior e retornar à coleta retilínea.	8 passível de k_c
T-07	Região do eixo como em (i)	Tocar o eixo de algum rolo: o ROSI deverá tocar a região do eixo de um rolo qualquer com a ponteira de toque. O toque ideal é feito com o vetor \vec{z}_s^+ alinhado ao vetor normal ao plano da face do eixo do rolo. A ponteira de toque possui uma complacência elástica; entretanto, a força de	8

		compressão no punho não pode passar de 1 N. Esta força será avaliada pelo sensor de força e torque localizado na flange do manipulador robótico.	
T-08	Rolo em fogo como em (d)	Identificar um rolo em chamas: o ROSI deverá identificar e indicar um rolo em chamas. A indicação deve ser marcando a localização do rolo defeituoso no momento da detecção em um mapa de preferência da equipe.	6
T-09	Região do cavalete como em (h)	Tocar o cavalete de algum rolo: o ROSI deverá tocar a região do cavalete de um rolo qualquer com a ponteira de toque. Não será considerado para um rolo em fogo. O toque ideal é feito com o vetor \vec{z}_s alinhado ao vetor normal ao plano do cavalete. A ponteira de toque possui uma complacência elástica; entretanto, a força de compressão no punho não pode passar de 1 N. Esta força será avaliada pelo sensor de força e torque localizado na flange do manipulador robótico.	5
T-10	Trajeto como em (f)	Coleta contínua desviando do obstáculo menor: enquanto realizando uma coleta retilínea ao lado do TC, o ROSI deverá desviar do obstáculo menor e retornar à coleta retilínea.	5 passível de k_c
T-11	Trajeto (e)	Coleta retilínea em região restrita: o robô deve passar por dentro da região restrita e realizar a coleta sem tocar qualquer estrutura.	5 passível de k_c
T-12	Trajeto como em (a)	Coleta retilínea no lado A: o robô deve realizar a coleta na seção sem obstáculos do lado A do TC.	3 passível de k_c
	-	Total Base	120

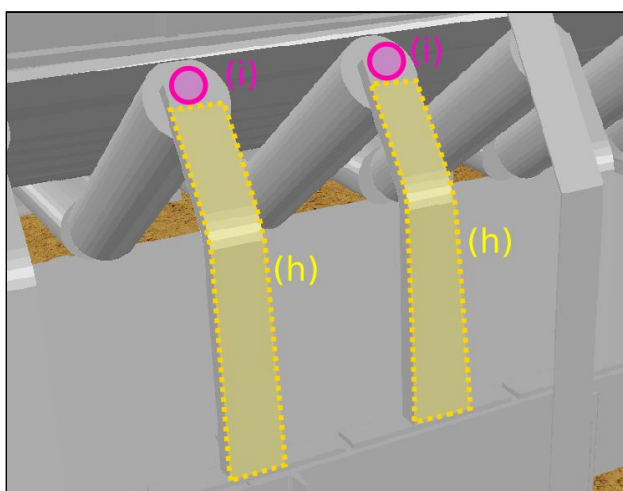
As Figuras a seguir destacam o local de interesse das tarefas, de acordo com a referência utilizada na Tabela 1. As imagens seguintes exibem os locais das tarefas **TABELA1.T-12**, **TABELA1.T-03**, **TABELA1.T-04** e **TABELA1.T-08** para realização no Lado A do Transportador de Correias.



As cenas subsequentes mostram em destaque as tarefas **TABELA1.T-03**, **TABELA1.T-04**, **TABELA1.T-11**, **TABELA1.T-06**, **TABELA1.T-10** e **TABELA1.T-08** para o Lado B do Transportador de Correias.



Para as tarefas **TABELA1.T-02**, **TABELA1.T-05**, **TABELA1.T-07** e **TABELA1.T-09** anteriores, estão destacadas a seguir as regiões de interesse (h) e (i) para inspeção.



A ordem de realização das tarefas não é sequencial, sendo que as equipes poderão escolher a estratégia de pontuação mais conveniente.

A pontuação de uma tarefa em específico é única. Os pontos de uma tarefa serão garantidos apenas sobre o melhor desempenho desta caso haja repetição. As penalizações serão computadas de maneira absoluta, *i.e.* cada infração cometida será pontuada com somatório sem limite superior.

O robô terá no máximo **30 minutos**, contados no **tempo de simulação**, para completar as tarefas. O tempo de simulação pode ser acompanhado na janela de simulação sob o título **simulation time**.

A Tabela 2 a seguir descreve as situações que poderão gerar penalizações às equipes.

Tabela 2 - situações de penalização

Código	Descrição	Penalização
F-01	Capotamento: ocorre quando qualquer estrutura do ROSI, salvo o sistema de locomoção, tocar o chão. Entende-se como sistema de locomoção as rodas e as sapatas da esteira de tração.	Fim da simulação
F-02	Inação: ocorre quando o deslocamento absoluto do robô é menor que 1 (um) metro por um período maior que 5 (cinco) minutos de simulação.	Fim da simulação
F-03	Saída do ambiente de simulação: o ambiente de simulação é finito. Se o robô ultrapassar a borda, saindo do ambiente, a simulação será encerrada.	Fim da simulação
F-04	Toque indevido do robô: choque do robô com a estrutura do transportador de correias, escada ou qualquer superfície não prevista em tarefas. Será aplicada cada vez que ocorrer contato indevido do ROSI com qualquer item do ambiente. Entende-se como contato indevido choques do manipulador, sensores, plataforma móvel ou braço com as superfícies adjacentes.	2
F-05	Compressão excessiva na ponteira de toque: penalização caso a força mensurada na flange do UR5 durante o toque ultrapasse 1 N. A força é medida no tópico <code>/ur5/forceTorqueSensorOutput</code> . Um toque é considerado enquanto a ponteira estiver em contato com a estrutura do TC.	5

As condições que levam à penalização de “*Fim de simulação*” implicam na finalização da simulação e soma da pontuação e penalização obtidas até a ocorrência da falha.

A Tabela 3 aborda a pontuação referente à qualidade de código, documentação e reprodutibilidade da solução apresentada. A pontuação do código só ocorrerá caso seja realizada ao menos 1 (uma) tarefa da Tabela 1.

Tabela 3 - pontuação do código

Código	Descrição	Pontuação
P-01	Legibilidade. Os códigos deverão estar bem estruturados, comentados e organizados. As funções devem ter descrição de variáveis de entrada e saída. Os nomes das variáveis devem ser inteligíveis e auto-explicativos. Números e variáveis “mágicas” deverão ser evitados.	Excelente: 10 Bom: 8 Regular: 5 Ruim: 0
P-02	Documentação. O README.md do repositório deve explicar de forma correta a estrutura do repositório, como instalar dependências e executar o pacote. Deverá ser implementado o wiki do repositório, indicando as funções implementadas, como usá-las, organização do pacote, tópicos publicados/subscritos e tipos de mensagens, relação com outros nós e etc.	Excelente: 15 Bom: 8 Regular: 3 Ruim: 0
P-03	Reprodutibilidade. O código deve ser facilmente executado em outra máquina qualquer com as especificações do desafio. O README.md deve fornecer informações precisas e sucintas para a instalação de dependências. Os pacotes ROS de terceiros devem estar recursivamente referenciados para clonagem. Deve ser fornecido um único arquivo launch para levantar todas as configurações e nós necessários da equipe.	Excelente: 5 Bom: 2 Ruim: 0
-	Total	30

A nota final de uma simulação é dada pela seguinte equação:

$$nota_s = \sum (k_b \cdot k_{c_i} \cdot pto_i) + b \cdot \sum pto_c - \sum (pen_i \cdot f_i)$$

Em que:

- $nota_s$ = nota da simulação;
- k_{c_i} = fator de multiplicação de sucesso da coleta durante a execução da tarefa i ;
- k_b = fator de multiplicação da completude da tarefa específica, estabelecido pela banca no intervalo $[0, 1]$;
- pto_i = pontuação base de inspeção para a tarefa i conforme Tabela 1;
- pen_i = penalização de inspeção i conforme indicado na Tabela 2;
- f_i = número de vezes em que a ação indevida i foi cometida;
- b = booleano que indica a execução de ao menos 1 (uma) tarefa de inspeção na simulação, tendo valor 1 para execução e 0 para não execução;
- pto_c = pontuação provinda da conformação dos códigos apresentados de acordo com a Tabela 3.

Estrutura do desafio

O desafio será dividido em **três etapas**: pré-avaliação, qualificatória/eliminatória e final. As datas importantes estão sintetizadas na Tabela 4, sendo referenciadas no texto por códigos, e.g. **TABELA4.C-XX**.

Não serão admitidos atrasos sob quaisquer circunstâncias, sendo que o não cumprimento de alguma data/horário especificados configura na eliminação da equipe.

Os cenários, modelo do ROSI e códigos do simulador utilizados nas etapas de avaliação serão exclusivamente os disponibilizados pela organização. As equipes não poderão, sob hipótese alguma, solicitar a avaliação em uma versão modificada do simulador.

Etapas

A presente seção descreve as etapas do desafio. As 2ª (Qualificatória/Eliminatória) e 3ª (Final) etapas consideram as pontuações e penalizações descritas na Seção **Tarefas de Inspeção, Pontuações e Penalizações**.

1ª Etapa - Pré-avaliação

Esta etapa visa conformar a estrutura de implementação das equipes com o sistema de avaliação do desafio, não se tratando de uma etapa qualificatória e/ou eliminatória.

A organização coletará os códigos das equipes no horário especificado em **TABELA4.C-03**. Os códigos serão avaliados quanto a:

- factibilidade de execução no simulador;
- factibilidade de instalação de dependências;
- estruturação como um único pacote do ROS (demais pacotes deverão estar sob esta estrutura);
- acesso do repositório da equipe pelo usuário Github oficial do desafio;
- presença de erros quaisquer que inviabilizam a execução do código;
- legibilidade de código (estruturação, comentários, etc.);
- catalogação de uso do código.

A organização irá avaliar os pontos acima e enviar um e-mail (**TABELA4.C-04**) para o capitão da equipe com a indicação de eventuais problemas encontrados.

Não é garantido que todos os erros sejam identificados pela organização. A mesma indicará apenas os problemas que inviabilizaram a execução do código das equipes nos computadores oficiais do desafio. Lembrem-se que é dever da equipe manter um código

funcional e reprodutível, não podendo a equipe alegar qualquer responsabilidade da organização por problemas na execução dos códigos durante a competição.

2ª Etapa - Qualificatória/Eliminatória

A partir da segunda etapa, será avaliada a solução das equipes para a inspeção autônoma de Transportadores de Correia. Os códigos das equipes serão coletados através de um *script* pontualmente em **TABELA4.C-05**. A organização irá executar **uma única vez** a solução de cada equipe em um **cenário idêntico** ao disponibilizado no repositório oficial. Um vídeo gerado da simulação será usado como base para o cálculo da nota final.

Nesta etapa, a comissão organizadora definirá os valores de k_c e k_b aplicáveis às pontuações base, bem como os demais critérios de avaliação. A pontuação por qualidade do código não será avaliada nesta etapa, fixando $b = 0$. O resultado da etapa será disponibilizado em **TABELA4.C-06**. As equipes serão classificadas de acordo com a nota obtida, da maior para a menor, sendo classificadas até **10 equipes** para a etapa final; as demais serão eliminadas do desafio. Caso duas ou mais equipes empatem em 10º lugar, todas passarão à etapa final.

3ª Etapa - Final

Os códigos serão coletados para a etapa final em **TABELA4.C-07**. A partir da coleta final, os repositórios das equipes deverão ser configurados como **público** até **TABELA4.C-08** sob pena de **eliminação** caso não feito.

A organização irá executar **três vezes** o código de cada equipe em um **cenário similar** ao disponível no repositório do desafio. As diferenças com respeito ao cenário original estarão restritas à disposição dos **obstáculos**, da **região restrita** e dos **rolos em chamas**.

Os vídeos gerados, em cada execução, serão disponibilizados às equipes em **TABELA4.C-09**. As equipes deverão escolher **1 (um) vídeo** que julguem ser o melhor desempenho de sua solução até **TABELA4.C-10**. Este vídeo será exibido para a comissão avaliadora na etapa final em **TABELA4.C-11**.

Nesta etapa, cada avaliador julgará individualmente o valor de k_c , k_b e f_i para cada equipe. Cabe a comissão organizadora definir os valores de b e pto_c . A nota final da etapa, $nota_f$, é computada pela média simples da nota $nota_{s,j}$ do avaliador j:

$$nota_f = \frac{1}{3} \sum_{j=1}^3 nota_{s,j}$$

A nota da 3ª Etapa **não é cumulativa** com a nota da 2ª Etapa.

Datas do Desafio

A Tabela a seguir apresenta as datas importantes do desafio:

Tabela 4 - das datas do desafio

Código da etapa	Descrição	Data (2019)
C-01	Período de inscrição	De 29/07 a 19/08
C-02	Data final para questionamento sobre o regulamento	26/08
C-03	Coleta dos códigos para a pré-avaliação	23h59m do dia 29/09
C-04	<i>Feedback</i> dos códigos	Até 02/10
C-05	Coleta do códigos para a etapa qualificatória	23h59m do dia 06/10
C-06	Resultado da etapa qualificatória	Até 13/10
C-07	Coleta dos códigos para a etapa final	23h59m do dia 20/10
C-08	Modificação dos repositórios para público	21/10
C-09	Envio dos vídeos para as equipes	Até 24/10
C-10	Decisão do vídeo a ser exibido na etapa final	Até 12h00m do dia 25/10
C-11	Etapa final do desafio	28/10

Classificação Final e Premiação

A classificação final das equipes será em função das notas finais decrescentes obtidas na 3ª Etapa. Caso haja empate, fica a critério do comitê avaliador votar pelo melhor desempenho.

A disponibilização da classificação final e ganhadores será no mesmo dia da etapa final. As informações serão divulgadas presencialmente no SBAI e também no Twitter oficial do desafio (endereço no final do regulamento).

Por poderem participar remotamente, a entrega das premiações será combinada com o orientador (caso houver) ou capitão, na falta de um orientador, das equipes

vencedoras. O prêmio total será entregue nominalmente ao orientador. Caso não haja orientador, o prêmio será então dado nominalmente ao capitão da equipe.

A premiação será no **formato de brindes** de valor total em função da colocação das equipes:

- 1º lugar: **R\$ 10.000,00** em prêmios;
- 2º lugar: **R\$ 7.000,00** em prêmios;
- 3º lugar: **R\$ 3.000,00** em prêmios.

Inscrições

Os participantes deverão inscrever suas equipes gratuitamente no ROSI Challenge através do formulário:

<https://forms.gle/2Zcy8dJjXGBqGNKS9>

Entende-se por participante de uma equipe todo membro ou orientador que faça parte desta. Membro é aquele que faz parte e não figura como orientador da equipe.

As equipes podem ser compostas de até **10 (dez)** membros, **mais 1** (um(a)) **orientador(a) (facultativo)**, podendo então haver até **11 participantes** por equipe. Como restrições:

- o(a) orientador(a) deve estar, obrigatoriamente, vinculado(a) a um instituto de pesquisa ou a uma instituição de ensino, como professor(a) e/ou pesquisador(a). O(A) orientador(a) deve, obrigatoriamente, possuir cidadania e residência brasileira;
- um mesmo(a) orientador(a) pode estar vinculado a mais de uma equipe;
- não é requerido aos membros estarem ligados a qualquer tipo de instituição;
- cada membro só pode participar de uma única equipe.

Cada equipe deverá designar um membro como capitão (capitã), responsável por interagir com a comissão organizadora. O **capitão responde por todos da equipe** em quaisquer situações. O(A) capitão (capitã) e orientador(a) devem, obrigatoriamente, possuir cidadania brasileira e residir no Brasil.

A formação da equipe indicada no momento da inscrição é inalterável. A única exceção está no desligamento voluntário de um membro, que deverá ser encaminhado de forma escrita e assinada pelo próprio à comissão organizadora através do e-mail oficial.

Deverá ser incluído como colaborador do repositório privado da equipe o usuário Github **rosichallenge** e enviado o link de convite no formulário de inscrição. A coleta das propostas para análise será realizada a partir deste usuário e será imprescindível sua inclusão como colaborador.

Portanto, cada equipe fornecerá no ato da inscrição:

- o nome da equipe* e link do convite* para acesso ao repositório privado no Github para o usuário **rosichallenge**;
- informações de cada membro e orientador(a): nome completo*, CPF*, e-mail*, telefone, instituição e grau de formação, link para currículo lattes e LinkedIn, ocupação e organização de atuação.

Os itens anteriormente destacados com (*) deverão ser informados obrigatoriamente para todos participantes da equipe.

As inscrições estarão abertas no período destacado na **TABELA4.C-01**, sendo vedada a participação de equipes com inscrição posterior a esta.

Antes de efetuar a inscrição, as equipes deverão conhecer o regulamento e certificar-se de que preenchem todos os requisitos exigidos.

As informações prestadas no formulário de inscrição serão de inteira responsabilidade das equipes, dispondo a Comissão do direito de excluir da competição aquela equipe que não preencher o formulário de forma completa e correta. Os dados fornecidos na inscrição serão utilizados exclusivamente no desafio, não sendo fornecidos à terceiros.

Demais Informações do Desafio

Seguem as informações gerais sobre o desafio.

Comissão organizadora e técnica

A comissão organizadora e técnica do ROSI Challenge é composta pelos membros:

- Filipe Rocha - Instituto Tecnológico Vale;
- Wagner Ferreira Andrade - Instituto Tecnológico Vale.

Comissão de apoio

A comissão organizadora agradece profundamente a todas as pessoas que contribuíram de alguma forma para a realização deste desafio:

- Prof. Sandoval Carneiro Jr, PhD - Instituto Tecnológico Vale (ITV);
- Jorge M. C. Fernandes - VALE;
- André Stanzani Franca - VALE;
- Fernando Lizarralde - PEE, COPPE, UFRJ;
- Amauri Coelho Ferraz - Instituto Tecnológico Vale (ITV) / Unicamp;
- Raphael Pereira Figueiredo da Silva - Escola Politécnica, UFRJ.

Comitê avaliador

O comitê para avaliação dos trabalhos será composto por três membros, observados os suplentes:

- 1º Avaliador: Agnaldo J. Rocha - UFOP;
- 2º Avaliador: Felipe Ribeiro - VALE;
- 3º Avaliador: Gustavo Freitas - UFMG;
- 1º Suplente: Gustavo Pessin - Instituto Tecnológico Vale (ITV);
- 2º Suplente: Filipe Rocha - Instituto Tecnológico Vale (ITV).

O comitê avaliador é soberano para decidir situações de avaliação não previstas neste regulamento.

Disposições gerais

Considera-se as seguintes disposições gerais:

- A solução desenvolvida por cada equipe deve ser única. É proibida a cópia de propostas entre equipes. A identificação de códigos similares levará à eliminação de todas as equipes envolvidas.
- A organização possui a autonomia de avaliar os pacotes/dependências de terceiros sugeridos pelas equipes para serem instalados. Caso julgue algum conteúdo como inapropriado - e.g. código malicioso, má referência de autor - a organização poderá solicitar um módulo substituto, ou mesmo eliminar a equipe;
- Não caberá recursos sobre as decisões da comissão organizadora ou comitê avaliador;
- Todos os participantes das equipes concordam que o capitão possui autonomia para resolver todas as questões do desafio diretamente com a organização.
- A organização irá entregar a premiação nominalmente ao orientador da equipe quando houver, ou nominalmente ao capitão na falta de um orientador. A organização não se responsabiliza ou toma parte sobre quaisquer questões envolvidas no uso ou distribuição deste recurso.
- Reserva-se à organização a autonomia de cancelar ou alterar qualquer ponto do desafio e/ou deste regulamento a qualquer momento sem qualquer prejuízo à organização. Eventuais modificações serão publicadas oficialmente no Twitter do desafio e catalogadas no repositório oficial.
- Reserva-se à organização a autonomia de julgar e decidir a qualquer momento sobre os casos omissos não previstos neste regulamento.
- É de inteira responsabilidade das equipes acompanhar a publicação de todos os comunicados no perfil oficial do desafio no Twitter.

Contato

As notícias deste desafio serão publicadas unicamente no Twitter oficial do desafio, [@rosichallenge](https://twitter.com/rosichallenge). Eventuais modificações do simulador serão indicadas no Twitter e listadas no [README.md](#) do repositório oficial.

As dúvidas quanto à utilização do simulador devem ser encaminhadas unicamente como issues no repositório do Github oficial do desafio: <https://github.com/filRocha/roSiChallenge-sbai2019>. Dúvidas deste tipo encaminhadas por e-mail não serão respondidas.

Por fim, dúvidas quanto ao regulamento ou à estrutura do desafio devem ser encaminhadas até **TABELA4.C-02**, através do capitão, ao e-mail oficial do desafio: rosichallenge@gmail.com.