## **Definição de arquitetura para módulos de suporte à deteção da atenção de pessoas na proximidade de robôs e geração de alarmes.**

* Módulo de deteção de pessoas
* Módulo de estimação de posição de pessoas no espaço do robô
* Módulo de deteção de contacto visual
* Módulo de estimação do nível de perigo relativo à presença de pessoas

### ***Dados utilizados***

* **[0A]** Fluxo de vídeo
* **[0B]** Informação de profundidade *(PCL)* **[opcional]**
* **[0C]** Posição corrente das partes do robô **[opcional]**
* **[0D]** Plano de movimento corrente **[opcional]**

### **[PplBBs] *Deteção de pessoas***

#### ***Resumo***

Este módulo converte o fluxo de vídeo num fluxo de posições de caixas englobantes *(bounding boxes)* de pessoas presentes na imagem. Caso **[0B]** é fornecido, então também fornece logo as profundidades a que as pessoas estão.

* ***Entradas:***
  + ***Imprescindível:***
* **[0A]** Fluxo de vídeo
* ***Opcional:***
* **[0B]** Informação de profundidade *(PCL)*
* ***Saídas:***
  + ***Garantido:***
    - **[1A]** Número de pessoas detetadas
    - **[1B]** Lista de caixas englobantes *(bounding boxes)* da cabeça*:* *uma por cada pessoa detetada*
  + ***Condicional ⇒*** *na ausência de* ***[0B]:***
    - **[1CN]** Lista de caixas englobantes *(bounding boxes)* do corpo inteiro *(incluindo cabeça):* *uma por cada pessoa detetada*
  + ***Condicional ⇒*** *se* ***[0B]*** *fornecido****:***
    - **[1CY]** Lista das profundidades fornecidas por **[0B]**: *uma por cada pessoa detetada*
* ***Parâmetros:***
  + ***Obrigatório:***
    - **[ign\_thresh]** Pelo menos um dos três: **[min\_bb\_head\_sz] ; [min\_bb\_full\_sz] ;** ou **[max\_dist]**.
      * *O parâmetro* ***[ign\_thresh]*** *serve para definir limiares para além dos quais pessoas são ignoradas em vez de serem detetadas. Isto é, se a pessoa apresenta características mais extremas do que pelo menos um dos limiares definidos, então a pessoa não figura em* ***[1B]*** *e* ***[1C\_]****, assim como não conta para o valor de* ***[1A]***
  + ***Opcional:***
    - **[min\_bb\_head\_size]** Tamanho mínimo da caixa englobante *(bounding box)* da cabeça da pessoa
  + ***Opção se se cumprir requisito ⇒*** *caso uma das saídas seja* ***[1CN]:***
    - **[min\_bb\_full\_size]** Tamanho mínimo da caixa englobante *(bounding box)* da pessoa
  + ***Opção se se cumprir requisito ⇒*** *caso uma das saídas seja* ***[1CY]:***
    - **[max\_dist]** Distância máxima da pessoa

### ***2.* [SetDgrZone] *Definição de zonas de perigo***

#### 

#### ***Resumo***

Este módulo define zonas de perigo concêntricas à volta do robô. O número de zonas e a distância entre a borda de cada uma são definidas como parâmetros. Caso o robô seja móvel, as zonas são calculadas com base no movimento planeado dele.

* ***Entradas:***
  + ***Opcional:***
* **[0C]** Posição corrente das partes do robô
* **[0D]** Plano de movimento corrente
* ***Saídas:***
  + ***Garantido:***
    - **[2A]** Posições das zonas de perigo
* ***Parâmetros:***
  + ***Obrigatório:***
    - **[n\_zones]** Número de zonas de perigo a definir
    - **[zone\_widths] “**Espessuras” das zonas
      * *Para cada zona, por “espessura“ entende-se a distância desde a fronteira com uma zona mais interior até a fronteira com uma zona mais exterior.*

### ***3.* [PplPosition] *Estimação da posição das pessoas***

#### 

#### ***Resumo***

Este módulo usa a informação fornecida por **[PplBBs]** para estimar a posição corrente das pessoas detestadas na imagem. Esta funcionalidade será realizada com base numa abordagem do tipo filtro probabilístico: Kalman Extendido; UKF; ou filtro de partículas.

* ***Entradas:***
  + ***Imprescindível:***
* **[0A]** Fluxo de vídeo
* **[1B]** Lista de caixas englobantes *(bounding boxes)* da cabeça*:* *uma por cada pessoa detetada*
* ***Condicional ⇒*** *se* ***[1CN]*** *fornecido****:***
* **[1CN]** Lista de caixas englobantes *(bounding boxes)* do corpo inteiro *(incluindo cabeça):* *uma por cada pessoa detetada*
* ***Condicional ⇒*** *se* ***[1CY]*** *fornecido****:***
* **[1CY]** Lista das profundidades fornecidas por **[0B]**: *uma por cada pessoa detetada*
* ***Saídas:***
  + ***Garantido:***
    - **[3A]** Lista de posições estimadas: *uma por cada pessoa detetada*
* ***Parâmetros:***
  + ***Obrigatório:***
    - **[camera\_matrix]** Matrizes de calibração da câmara

### ***4.* [PplLooked] *Deteção de contacto visual***

#### ***Resumo***

AAAAAAAAA

* ***Entradas:***
  + ***Imprescindível:***
    - **[0A]** Fluxo de vídeo
    - **[1B]** Lista de caixas englobantes *(bounding boxes)* da cabeça*:* *uma por cada pessoa detetada*
    - **[4A]** Lista de tempos que passaram desde a pessoa ter olhado

***Parâmetros:***

***Saída: verdadeiro/falso, para (id) cada pessoa***

### ***4. Estimação do nível de perigo***

* ***Entradas:***
  + imagem, pessoas e informação de contacto visual

***Parâmetros:*** Zonas associadas a cada nível de perigo

***Saída: Etiqueta para cada pessoa do nível de perigo***

*Este módulo fará uma estimação do nível de perigo com base na posição da pessoa e zona em que se encontra e com base no tempo que decorreu desde o último contacto visual estabelecido com o robô. Numa zona mais próxima, o perigo é muito elevado, numa zona intermédio o nível de perigo depende de quanto tempo passou desde o último contacto visual com o robô.*

*Temos casos diferentes para robôs móveis e manipuladores fixos dado que o espaço onde os mesmos evolvem é dinâmico nos primeiros, sendo estático para os segundos.*

*Parte-se do princípio que a consciência da presença ou aproximação do robô decresce rapidamente com o tempo desde o último contacto visual com o mesmo.*

*(podemos descrever aqui melhor as situações para robôs móveis e manipuladores estáticos.)*

De acordo com a severidade dos nível de perigo e contexto da instalação poderão ser acionados diversos mecanismos à escolha do “programador”: alerta visual, alerta visual + auditivo, alertas + redução da velocidade de operação do robô, paragem do robô.

#### ***Objectivo***

AAAAA

***Metodologia***

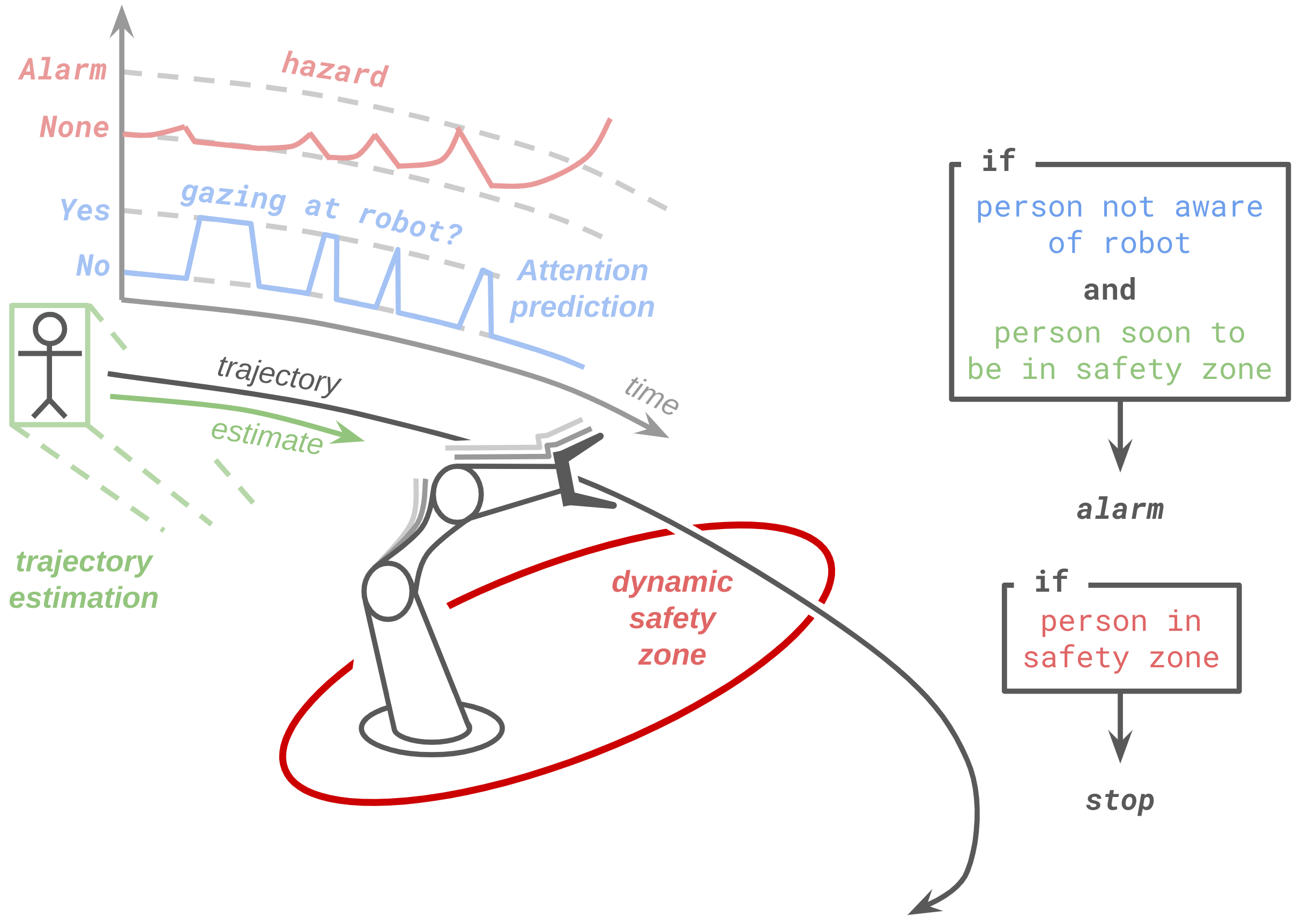
AAAAA

## **Dynamic safety zones sensitive to cobot awareness in human workers**

**Figure 1** shows a diagram of the intended architecture. It consists of three components that will work in tandem upon the appearance of a person in the robot’s field of view. Firstly, a dynamic***safety zone*** is defined around the cobot: unlike in SSM, the safety zone here is in close proximity to the cobot and changes shape in accordance with its planned position. The second component is a ***trajectory estimation*** module: as the person moves, the trajectory they take is inferred using the cobot’s vision, and the path taken is used to estimate the person’s future trajectory. Finally, the third component is an ***attention prediction*** module: the person’s timewise gaze pattern is captured using the cobot’s vision, and then used to determine if the person is currently aware of what the cobot is doing.

These three components will work in tandem as follows. The person cannot enter the safety zone: if they do, the cobot stops. However, if the person is aware of the cobot’s kinesics, they will likely not enter the safety zone, even if their current trajectory suggests otherwise. As such, the attention prediction and trajectory estimation modules work together: if the person does not appear to be aware of the cobot, and their estimated future trajectory crosses the safety zone, then the cobot issues an alarm to alert the person. Otherwise, the cobot continues working.

The advantage of this architecture over the current ISO standards is that it allows for a smaller stopping area surrounding the cobot, making the occurrence of cobot safety stops less common. On top of allowing the cobot to do its job, it will also increase the viability of HRC tasks, allowing for the introduction of cobot assistance to the more repetitive aspects of otherwise dynamic human tasks.



***Figure 1:*** *A diagram of the intended architecture. It consists of three components, each of them color-coded. The dynamic safety zone is highlighted in red; the components of the trajectory estimation are highlighted in green; and the components of the attention prediction are coded in blue.*

The architecture was described above for the case of a stationary cobot and a moving human: this is the most likely scenario in current workspaces. But as Industry 4.0 evolves, and HRCp / HRC become more common, increasingly likely will also be the opposite scenario: a mobile cobot and a human at a work station. The same three components of the architecture are applicable there, although the behavior has to be adjusted *(namely having the cobot adjust its own plan instead of issuing alarms and stopping)*, and the algorithms need to be far more robust in order to cover a more broad range of conditions.

Below is a more detailed description of each of the three components.

### ***Dynamic safety zone***

This component of the architecture defines a union of narrow curved volumes around the current position of the cobot’s manipulator and end-effector, as well as around their planned future positions within a certain time-frame. Namely, if the cobot is planning to work with objects entirely to its north, the safety zone to its south would be set narrower, but if the cobot then plans to transition to working on something to its west, then its safety zone would adjust in real time, moving away from the north toward the west.

The procedure for calculating the safe zone would be entirely based on the planned kinesics, so that no prediction or AI would be required for its calculation.

### ***Trajectory estimation***

This component will estimate the trajectory of the person based on RGB video footage from the vantage point of the cobot. A cartesian coordinate representation of the trajectory will be obtained by converting the inferred spherical coordinate position, where the person’s distance to the cobot is estimated by comparing the size they take up in successive frames; together with assumptions of reasonable human dimensions and proportions; as well as assumed smoothness in velocity vector variations.

As described above, the procedure is entirely a model-based multivariate regression. Parallax is not used, so that no image segmentation or complex image backgrounds are necessary; and DNN methods are eschewed so as to obviate large training datasets. The goal is to make the module as computationally cheap as possible, so ML methods will be applied to the regression task only if less computationally expensive statistical methods are deemed unreliable.

### ***Attention prediction***

This component will predict the degree to which the person is aware of the cobot’s kinesics. To do this, it will use RGB video footage to classify if the person is looking / not looking at the cobot over multiple frames. Depending on if the person is paying attention to the cobot or not, the pattern of when they look at it will differ. As such, these timewise patterns of binary classifications will be used to determine if the person is aware of the cobot.

The attention prediction module is reliant on DNNs to perform the gaze classification. Mutual gaze classifiers already exist and work reliably, but this case must account for people gazing at the manipulator as well, a more niche application, so that acquiring a reliable dataset may be a challenge. Possible approaches are producing a custom training dataset; repurposing an existing dataset *(e.g. pedestrians crossing a road in self-driving car training data)*; or even generating a life-accurate dataset based on a studied model of attentional behavior.

**Minimum requirements**

The minimum requirements are listed below according to the components to which they pertain.

* ***Dynamic safety zone:***
  + Accurate readings of speeds and positions of the different parts of the manipulator:
    - *This is needed in order to be able to accurately define the narrow volumes used to generate the safety zone.*
* ***Trajectory estimation:***
  + RGB camera mounted outside of the manipulator / end-effector portion of the cobot:
    - *If all cameras are mounted on the cobot’s moving parts, then an approaching person will only occasionally be captured in the frame. This may be sufficient for obtaining a rough estimate of the person’s trajectory, but the estimate will become increasingly inaccurate the longer the person is out-of-frame.*
  + Accurate knowledge of the same RGB camera’s azimuth:
    - *Either by having it be panoramic / 360º, or by having it rotate. The azimuth is one of the three spherical coordinates, so it is necessary for obtaining a cartesian trajectory.*
* ***Attention prediction***
  + Either a thin client or a computer capable of running a pre-trained DNN.
    - *Necessary for running the DNN that will classify if the person is aware of the cobot.*

A more robust sensor / computational suite will naturally provide improved reliability. Namely, the use of two RGB cameras or an RGBD camera will greatly improve trajectory estimation, while increased computing power would improve the efficiency of the attention prediction module.

**Equipment needed**

We have access to two cobots at ISR, as well as an industrial workspace lab at Polo 1. Although this is sufficient for some live testing, it is not a good predictor of how agnostic our architecture is for different cobot models, integrations, or applications. The logistics of performing live tests at a real plant are risky and expensive due to the downtime, besides not giving any information on yet-to-be-integrated cobots. As such, a workspace simulation software can be very useful. The Professional edition of RoboDK provides many of the features required for testing prototypes in this project: *provide examples and explain why some cheaper / open-source solution won’t work.*

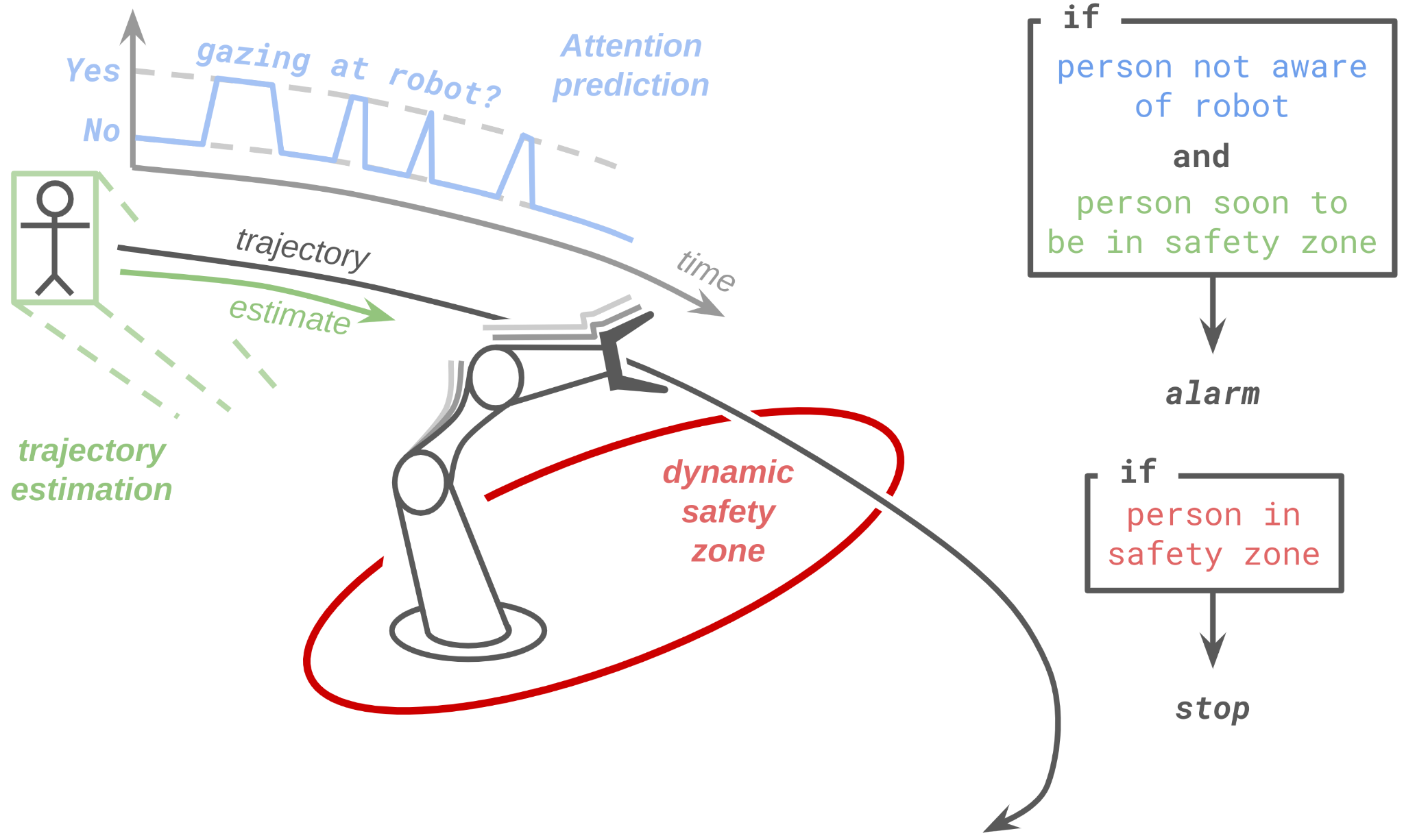
*RoboDK offers a broad library of robots not only a specific brand of robots unlike RobotStudio by ABB and MotoSim by FANUC. comparing with open-source solutions for instance Webots and Gazebo-ROS which are more suitable for research purposes and not industrial robots, RoboDK offers less steeper learning curve with less complication for users.*

Additionally, the attention prediction module will require a dataset with known attentive and known unattentive gaze patterns. Even if using a pre-existing dataset, it would be necessary to arrange the datapoints by gaze pattern. As such, the gaze pattern must be known first. It can be identified by simulating a workspace and then using a VR headset with gaze tracking to study the gaze patterns. Although ISR has access to several VR headsets, it does not have any with gaze tracking. The *HTC VIVE Pro Eye* provides such features, so it would be useful for the project.

To summarize, this project requires two pieces of equipment:

* RoboDK Professional
* *HTC VIVE Pro Eye*

### ***Old stuff***



### ***Deteção de pessoas***

* ***Entradas (I):***
  + ***Imprescindível:***
* **[1.I1] Fluxo de vídeo:** obrigatório
* ***Opcional:***
* **[1.I2] Informação de profundidade (PCL):** opcional

### ***Deteção de pessoas***

* ***Entradas (I):***
  + ***Imprescindível:***
* **[1.I1]** Fluxo de vídeo
* ***Opcional:***
* **[1.I2]** Informação de profundidade *(PCL)*
* ***Saída (O):***
  + ***Garantido:***
  + ***Condicional:***
  + Número de pessoas detetadas,
  + Caixas englobantes por cada pessoa *(bounding boxes)*
    - Informação de profundidade associada a cada pessoa *(caso PCL fornecido)*
* ***Parâmetros:***
  + Limiar para além do qual pessoas são ignoradas. ***Duas opções:***
    - Tamanho mínimo permitido para a caixa englobante da pessoa / cabeça da pessoa
    - Distância máxima da pessoa *(caso disponível)*
  + *distância máxima, ou tamanho mínimo da caixa englobante da pessoa ou da cabeça a partir da/o qual deve ser reportada a presença de uma pessoa.*