Learning Collaborative Impedance-Based Robot Behaviors

Leonel Rozo and Sylvain Calinon and Darwin Caldwell
Department of Advanced Robotics - Istituto Italiano di Tecnologia (Via Morego, 30. Genoa, Italy)

Pablo Jiménez and Carme Torras
Institut de Robòtica i Informàtica Industrial CSIC-UPC (Carrer Llorens i Artigas, 4-6. Barcelona, Spain)

Abstract
Research in learning from demonstration has focused on transferring movements from humans to robots. However, a need is arising for robots that do not just replicate the task on their own, but that also interact with humans in a safe and natural way to accomplish tasks cooperatively. Robots with variable impedance capabilities opens the door to new challenging applications, where the learning algorithms must be extended by encapsulating force and vision information. In this paper we propose a framework to transfer impedance-based behaviors to a torque-controlled robot by kinesthetic teaching. The proposed model encodes the examples as a task-parameterized statistical dynamical system, where the robot impedance is shaped by estimating virtual stiffness matrices from the set of demonstrations. A collaborative assembly task is used as testbed. The results show that the model can be used to modify the robot impedance along task execution to facilitate the collaboration, by triggering stiff and compliant behaviors in an on-line manner to adapt to the user’s actions.

1 Introduction
Over the last decade Robotics is addressing new challenging problems to bring robots closer to humans, as computers have now become part of our everyday life. It is envisaged that robots should collaborate with humans to perform a large variety of tasks more easily, faster and in a safer way. To accomplish this goal, robots must be endowed with learning capabilities allowing them to acquire new knowledge from examples given by a human or through their own experience. Learning from demonstration (LfD) is a natural way to transfer knowledge to robots from human examples (Billard et al. 2008). Most works have focused on developing learning algorithms to encode trajectories using vision or optical systems to capture the teacher demonstrations, see e.g., (Calinon, Sardellitti, and Caldwell 2010). Nevertheless, the new variable impedance capabilities of recent robotic arms (Rooks 2006; Albu-Schäffer et al. 2007) demand to reformulate these methods in order to exploit their new control schemes in performing more complex tasks.

In this line, physical human-robot interaction (pHRI) has arisen a lot of interest recently, with the two challenging aspects of impedance control and haptic communication. On the one hand, an increasing effort has been devoted to exploiting the advantages provided by the impedance-based control of robots (Hogan 1985). On the other hand, impedance in humans has also been studied with the aim of gaining in-depth knowledge of the roles of the muscles, tendons, brain and spinal cord in modulating impedance when we interact with the environment (Burdet et al. 2001; Gomi and Kawato 1996). Also, efforts have been devoted to mimic human impedance with robots (Ganesh et al. 2010). In this context, it is desirable to transfer impedance-based behaviors from humans to robots (Kalakrishnan et al. 2011; Kronander and Billard 2012). Evrard and Kheddar (2009) tackled the problem of setting the robot’s role in a collaborative lifting task by modifying the impedance controller parameters. The approach was then extended to let the robot learn this behavior autonomously from human demonstrations through probabilistic approaches (Calinon et al. 2009; Gribovskaya, Kheddar, and Billard 2011).

The second challenge in pHRI is the haptic communica-
tion between the partners. When physical interaction takes place, forces-torques sensed by the partners constitute a very rich and valuable communication channel which is used to recognize and/or predict intentions as well as to determine the role of each participant in the task (Evrand and Kheddar 2009; Groten et al. 2013; Reed and Peshkin 2008). The challenges from the robot side are to distinguish the haptic signals related to the task from those communicating intentions, and to anticipate the human actions using information about the dynamics of the task and the force-based perceptions generated along it (Thobbi, Gu, and Sheng 2011).

In this paper, we are concerned with learning collaborative impedance-based skills\(^1\) using visual and haptic information, where the robot behavior is conditioned by task variables as well as by its haptic perceptions. We thus require a model that can learn impedance behaviors from a set demonstrations, where the model is modulated by haptic and vision information. The proposed approach is tested in a collaborative assembly of a wooden IKEA table. In the learning phase two humans perform the task, where one is kinesthetically\(^2\) guiding the robot to demonstrate the robot’s role. The compliance behavior of the person changes to allow his/her partner to perform his/her corresponding sub-task more easily (see Figure 1). During reproduction, the robot replaces the user holding the table by automatically estimating the levels and shape of stiffness ellipsoid required during the interaction.

The remainder of the paper is organized as follows: Section 2 explains the learning algorithm and the compliance level estimation method. Section 3 describes the collaborative task and the experimental setting. Results are shown and analyzed in Section 4. Finally, conclusions and future work are presented in Section 5.

2 Proposed approach

We extend the task-parameterized movement learning approach recently proposed in Calinon et al. (2012) to a task-parameterized impedance learning problem. This approach relies on a statistical representation of dynamical systems that can be modulated with respect to task variables represented as candidate frames of reference. The model is here extended to force-based impedance behaviors requiring to adapt the stiffness of virtual springs in Cartesian space driving the robot’s behavior.

2.1 Task-parametrized Gaussian mixture model

When robots manipulate objects, their movements may largely depend on the given goals and object poses, which can be defined through reference frames. Namely, the robot motion is conditioned by a set of task variables representing the coordinate systems of relevant frames of reference. For generalization purposes, it is desirable to have a model enclosing different movements as a function of the these variables, instead of representing each one with a different model. The proposed approach relies on Gaussian product properties to modulate the centers and covariance matrices of a Gaussian mixture model (GMM). The advantages of this approach compared to other task-parameterized models such as the parametric Hidden Markov Model (PHMM) (Wilson and Bobick 1999) are discussed in Calinon et al. (2012).

Formally, each demonstration \(m \in \{1, \ldots, M\}\) contains \(T_m\) datapoints forming a dataset of \(N\) datapoints \((\xi_n)_{n=1}^N\) with \(N = \sum_{m} T_m\). Each \(\xi_n \in \mathbb{R}^D\) is associated with task variables \((A_{n,j}, b_{n,j})_{j=1}^{N_P}\) representing \(N_P\) candidate frames of reference, with transformation matrices \(A_{n,j}\), and offset position vectors \(b_{n,j}\). \(D\) is the datapoint dimensionality, and the indexes \(n\) and \(j\) represent the time step and the candidate frame, respectively.

The parameters of the model are \((\pi_i, Z_{i,j}^e, Z_{i,j}^v)\), representing respectively the mixing coefficients, centers and covariances matrices for each frame \(j\) and mixture component \(i\). With this model, for an observation of frames at iteration \(n\), the resulting center \(\mu_{n,i}\) and covariance matrix \(\Sigma_{n,i}\) of each component \(i\) are computed as products of linearly transformed Gaussians

\[
N(\mu_{n,i}; \Sigma_{n,i}) = \prod_{j=1}^{N_P} N(A_{n,j} Z_{i,j}^e + b_{n,j}, A_{n,j} Z_{i,j}^v A_{n,j}^\top).
\]

By using the product property of normal distributions, the above equation is computed as

\[
\mu_{n,i} = \sum_{j=1}^{N_P} (A_{n,j} Z_{i,j}^e A_{n,j}^\top)^{-1} (A_{n,j} Z_{i,j}^e + b_{n,j}),
\]

\[
\Sigma_{n,i} = \left( \sum_{j=1}^{N_P} (A_{n,j} Z_{i,j}^v A_{n,j}^\top)^{-1} \right)^{-1}.
\]

The parameters of the model are iteratively estimated with the following EM procedure. In the E-step, (1) are used as temporary Gaussian parameters to compute the likelihood.

E-step:

\[
h_{n,i} = \frac{\pi_i N(\xi_n | \mu_{n,i}; \Sigma_{n,i})}{\sum_k \pi_k N(\xi_n | \mu_{n,k}; \Sigma_{n,k})}.
\]

M-step:

\[
\pi_i = \frac{\sum_n h_{n,i}}{N}, \quad Z_{i,j}^e = \frac{\sum_n h_{n,i} A_{n,j}^\top (\xi_n - b_{n,j})}{\sum_n h_{n,i}},
\]
The model parameters are initialized with a $k$-means procedure, modified by following a similar task-parametrized structure. Model selection is compatible with the techniques employed in standard GMM (Bayesian information criterion, Dirichlet process, etc.).

One novelty with respect to (Calinon et al. 2012) is that we augment the model with virtual stiffness matrices $K_i^P$ associated to each component $i$, which will be estimated as explained in Section 2.2. Thus, the complete set of parameters of the model is $\{\pi_n, Z_c^{\pi}, Z_{c,i}^{\pi}, \mu_{n,i}, \Sigma_{n,i}, K_i^P\}_{i=1}^{N,i}$. Such extension allows us to apply the learning model to impedance-based behaviors transfer. Note that the variables of the task are obtained from the position and orientation of a set of candidate frames to learn the task. In our experimental setup, the table legs and robot frames define variables in the collaborative assembly task (described later in Section 3).

Fig. 3 illustrates the approach with a simple example. (a) shows the demonstrations where the robot behaves compliantly when another object (the green triangle) is far from its end-effector, and becomes stiff when the object approaches it with a specific orientation. (b) displays the two phases of the task, where the robot motion is driven by a set of virtual springs connected to the center of the model’s Gaussians. The mean and covariance vary according to the task variables (i.e., the object and robot frames), and the influence of each model component (see Eq. (2)) determines how compliantly the robot behave.

### 2.2 Stiffness estimation

Several approaches have been proposed to estimate from collected data the stiffness and damping parameters to control robots. Erickson, Weber, and Sharf (2003) compared four different methods to estimate the robot impedance based on signal processing, adaptive control and recursive least squares. Flacco and Luca (2011) estimated the nonlinear stiffness of robot joints with flexible transmissions by using dynamic residual signals along with least-squares and regressor-based techniques. From a different perspective, a LfD approach was proposed in Calinon, Sardelli, and Caldwell (2010) to find a stiffness matrix using variability information extracted from training data in the form of a GMM, where the stiffness matrix is estimated from the inverse of the observed covariance in the position space. Similarly, Lee and Ott (2011) used variability encoded in the components of an HMM to define a motion refinement tube that permits a deviation from nominal trajectories for kinesthetic corrections by controlling the stiffness value at the robot joints level.

Here, we obtain an approximation through an algebraic closed-form solution to find the closest symmetric positive semi-definite stiffness matrix of a weighted least-squares (WLS) estimation. A stiffness matrix $K_i^P$ is estimated for each component $i$, by assuming that the robot behavior is driven by a set of virtual springs (similar to Fig. 3)

$$F_n = \sum_{i=1}^{N,K} h_{n,i} \left[ K_i^P (\mu_{n,i}^x - x_n) \right],$$

where $F_n$, $\mu_{n,i}^x$, and $x_n$ are respectively the sensed force, the positional part of the Gaussians’ centers in the model (see Eq. (4)), and the robot’s end-effector position at time step $n$.

WLS is used to compute a first estimate $\hat{K}_i^P = \left( [X_i^T W_i X_i]^{-1} X_i^T W_i F \right)$ of the stiffness matrices by concatenating all the $N$ datapoints in matrices $X_i = \left( [\mu_{n,i}^x - x_1], \ldots, [\mu_{N,i}^x - x_N] \right)^T$ and $F$, with a weighting matrix $W_i = \text{diag}([h_{1,i}, h_{2,i}, \ldots, h_{N,i}])$ (see Eq. (2)). Such estimate does not necessarily comply with the symmetric positive semi-definite constraints of a stiffness matrix. Therefore, we resort to the formulation presented in (Higham 1988), to compute $K_i^P$ as the nearest symmetric positive semi-definite (SPSD) matrix to $\hat{K}_i^P$ according to the Frobenius norm, computed as

$$K_i^P = \frac{B + H}{2}, \quad B = \frac{\hat{K}_i^P + (\hat{K}_i^P)^T}{2}, \quad H = V \Sigma V^T.$$

}\[\text{Figure 3: Simplified impedance behavior learning. (a) 3 different demonstrations showing compliant and stiff phases. The black line is the robot’s trajectory. (b) Reproduction of the task, where the robot’s behavior is governed by a 2-states model with virtual springs connected to the Gaussians’ centers. Dark ellipses and thick-line springs represent an activated Gaussian. The candidate frames are displayed in red color.}\]
Table 1: Learning and reproduction phases.

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>- Determine $N_P$ (number of frames or task variables)</td>
<td>- Determine $N_K$ (number of components of the model)</td>
<td>- Find $K^{F}_{i}$ for each virtual spring by using Eq. (6)</td>
</tr>
<tr>
<td>- $\forall n \in {1, \ldots, N}$, collect $\xi_n$ and ${A_{n,j}, b_{n,j}}$ for each virtual spring</td>
<td>- Use Eq. (3) to learn ${\pi_i, {Z^\nu_{i,j}, Z^{\tau \nu}<em>{i,j}}</em>{j=1}^{N_P}}_{i=1}^{N_K}$ through Eq. (1)</td>
<td>- Compute activation weights $\beta_{n,i}$ using Eq. (2)</td>
</tr>
<tr>
<td>4. Reproduction (for each time step $n$)</td>
<td>- Collect $\xi_n$ and ${A_{n,j}, b_{n,j}}$ for each virtual spring</td>
<td>- Apply the force command computed from Eq. (5)</td>
</tr>
<tr>
<td>- Determine $\pi_i$</td>
<td>- Find $\pi_i$ and ${Z^\nu_{i,j}, Z^{\tau \nu}_{i,j}}$ through Eq. (1)</td>
<td></td>
</tr>
<tr>
<td>- Use Eq. (3) to learn ${\pi_i, {Z^\nu_{i,j}, Z^{\tau \nu}<em>{i,j}}</em>{j=1}^{N_P}}_{i=1}^{N_K}$ through Eq. (1)</td>
<td>- Apply the force command computed from Eq. (5)</td>
<td></td>
</tr>
<tr>
<td>- Collect $\xi_n$ and ${A_{n,j}, b_{n,j}}$ for each virtual spring</td>
<td>- Use Eq. (3) to learn ${\pi_i, {Z^\nu_{i,j}, Z^{\tau \nu}<em>{i,j}}</em>{j=1}^{N_P}}_{i=1}^{N_K}$ through Eq. (1)</td>
<td></td>
</tr>
</tbody>
</table>

$H$ is the symmetric polar factor which can be found from the singular value decomposition of $B$, namely, $B = U \Sigma V^T$. Table 1 summarizes the learning and estimation processes.

3 Collaborative assembly task

We consider a human-robot collaborative task where the robot’s role is to hold a wooden table while the user’s role is to screw the four legs to it. Fig. 2 presents an example of assembly instructions that can be found in “do it yourself” furniture catalog. Here, two small tables require specific sequences of force and movement to get assembled. Learning such specificities is required for an efficient collaborative assembly. Instead of manually programming those specificities for each item, we would like the robot to extract those automatically from a set of demonstrations provided by two users collaborating together to assemble the different parts of the table (see Fig. 1). After learning, the task can be reproduced by a single user, with the robot partner interacting appropriately with respect to the preferences of the user and the specificities of the item being assembled. We thus do not need to provide the robot with information about the points of assembly, the different options, orientation of table legs, etc. The robot instead learns these specificities from demonstrations.

3.1 Experimental setup

We use a KUKA lightweight 7-DoF robot (LWR) (Albu-Schäffer et al. 2007), with the Fast-Research Interface (Schreiber, Stemmer, and Bischoff 2010), by using a Cartesian impedance controller defined by

$$\tau_d = J^T F_d + V (\kappa_{d}^{V}) + f(q, \dot{q}, \ddot{q}),$$

where $J$ is the Jacobian of the robot, $\tau_d$ is the desired torque, $F_d$ the desired force computed from the resulting set of virtual springs (Eq. 5), $V$ is a damping function with desired damping values $\kappa_{d}^{V}$ and $f(q, \dot{q}, \ddot{q})$ the dynamic model of the robot.\footnote{Note that we only control the Cartesian position of the robot while the rotational DOF are set to be fixed during the reproduction phase.}

The position and orientation of the table legs are tracked with a marker-based NaturalPoint OptiTrack motion capture system, composed of 12 cameras working at a rate of 30 fps. A transformation matrix is computed to represent the leg configuration in the fixed robot frame $O_{R}$, from which $b_{n}^{R}$ and $A_{n}^{R}$ define the Cartesian position and the orientation of the leg as a rotation matrix, respectively. During both demonstration and reproduction phases, $\{A_{n}^{R}, b_{n}^{R}\}$ are recorded at each time step $n$ to determine the task variables. Lastly, the other candidate frame $\{A_{n}^{S}, b_{n}^{S}\}$ define the robot’s fixed frame of reference.\footnote{A 3D coordinate frame is replicated for the variables $x$, $F$ and $T$, the offset is only set to $x$.}

Figure 4: Reproduction results at different phases of the interaction. For each graph, the projection of the model’s Gaussians in the tool’s frame (as ellipses) is shown on the left, while the right part shows the trace of the resulting stiffness matrix. The black dotted line represents the leg’s trajectory, the table is shown in yellow with its 4 threads and the brown cross corresponds to the current position of the leg.

The position and orientation of the table legs are tracked with a marker-based NaturalPoint OptiTrack motion capture system, composed of 12 cameras working at a rate of 30 fps. A transformation matrix is computed to represent the leg configuration in the fixed robot frame $O_{R}$, from which $b_{n}^{R}$ and $A_{n}^{R}$ define the Cartesian position and the orientation of the leg as a rotation matrix, respectively. During both demonstration and reproduction phases, $\{A_{n}^{R}, b_{n}^{R}\}$ are recorded at each time step $n$ to determine the task variables. Lastly, the other candidate frame $\{A_{n}^{S}, b_{n}^{S}\}$ define the robot’s fixed frame of reference.\footnote{A 3D coordinate frame is replicated for the variables $x$, $F$ and $T$, the offset is only set to $x$.}

The collaborative scenario consists of screwing the legs at the four corresponding positions on the table. $D_{T}$ is first compliant to allow $D_{L}$ to move the table freely (compliant phase) until comfortable position and orientation are found for the work to be performed next. When $D_{L}$ grasps a leg and starts inserting it into the screw thread in the table, $D_{T}$ adopts a stiff posture, holding the table to facilitate $D_{L}$’s part
of the task (stiff phase).

Note that the combination of vision and haptic information is fundamental for this task. If only vision is used, the robot cannot distinguish the phase during which the user aligns the screw with the thread. Here, $D_T$ should instead regulate its stiffness in accordance with the sensed force pattern. If $D_T$’s behavior was based only on forces, the collaboration could fail because $D_T$ could not distinguish which forces correspond to interaction with $D_L$ and which are produced by the screwing actions. This can be problematic because these patterns might be similar in some situations. Both perception channels are thus needed to learn how the impedance behavior should be shaped.

4 Results

A model of five components ($N_K = 5$) was trained with sixteen demonstrations (i.e., each leg is assembled four times to its corresponding thread with specific vision and force patterns). The resulting model automatically discovered four stiff components corresponding to the four screwing phases, with the remaining component representing the compliant behavior. Each “stiff component” is characterized by the force-torque pattern and the relative position of the leg with respect to the robot tool frame, which are different for each leg. The “compliant component” encodes the remaining points in the data space, i.e., the interaction forces-torques as well as the varying robot end-effector and leg positions. Fig. 4 shows that the Gaussian corresponding to the compliant phase is already spatially distinguishable from the Gaussians encoding the stiff behaviors during the screwing processes (four in this case). Note that the Gaussians in the model representing the stiff phases show an elongated shape changing its orientation during the task. Such type of time-varying information encapsulated in the covariance matrices cannot be encoded by using the classic PHMM (no covariance parameterization).

Once the model is learned, the stiffness estimation is carried out as described in Section 2.2. In this experiment a stiffness matrix is locally associated with each component in the model, describing a virtual spring connected to the center of the Gaussian. During reproduction, a force command is estimated as a combination of the virtual springs (see Eq. 5).

The proposed approach was compared to the stiffness estimation procedure based on the inverse of the observed covariance (Calinon, Sardellitti, and Caldwell 2010) (see Fig. 5), by computing the inverse of the sub-matrix $\Sigma_{n,i}$ for each Gaussian $i$ at each time step $n$. A weighted average stiffness $K_{inv}$ is then calculated. This is compared to $K_{pnh}$ obtained as described in Section 2.2. With our training set, both approaches estimate the different stiffness levels appropriately. However, the estimate of Calinon, Sardellitti, and Caldwell (2010) has the disadvantage that it takes only into account the positional information from the data, whose variability can sometimes be too weak if only a few demonstrated scenarios are considered. In the experiment, the users covered various portions of the workspace. In a more realistic scenario, the users might not be aware of this scaffolding teaching procedure, and a smaller number of datapoints might be acquired. In such situation, variability information may not always be sufficient to estimate stiffness information. In contrast, the approach that we adopt in this paper does take into consideration the haptic inputs in the estimation process. Fig. 5 displays the trace of the estimated stiffness matrices for each Gaussian, comparing the results obtained by both approaches. The ratio between the stiff and compliant values (computed from the matrix traces) is higher using the proposed approach (10.12 as average) than those obtained from the approach based on position variability (5.19 as average), which allows a better clamping of the robot stiffness considering the obtained maximum and minimum values. This indicates that the difference between the compliant and stiff levels is more pronounced when the estimation process is based on the haptic data.

Previous experiments with force and position data have shown that it was often more challenging to exploit force recordings than positions in practice, mostly due to the lower accuracy of the sensors and disturbing forces (Rozo, Jiménez, and Torras 2013). However, a small variability has been observed for force signals in the second phase of the
task where the leg is screwed to the table. We attribute this low variability to two main factors: 1) the force signals pre-processing and 2) the fact that our experiment protocol facilitated the collaborative task to the users, which resulted in high consistency in the demonstrations. The two users demonstrated the collaborative assembly without talking, but by using their hands and their eyes. During reproduction, the robot however only had access to a limited subset of haptic and vision cues. For example, the users could observe the changes of posture and even the gaze of each other (Strabala et al. 2012; Mutlu, Terrell, and Huang 2013), which likely provided additional cues to jointly agree about the change of stiffness behaviors.

We tested the reproduction and generalization capabilities of the system by carrying out the assembly process for all the legs. In Fig. 6, we can observe how the compliant component (purple dotted line) is influential during the first half of the reproduction, dominating the other components. After this, the robot becomes stiff, with specific patterns depending on which leg is being screwed. This means that not all the components influence the robot impedance at the stiff phase, but mostly the Gaussian encoding the stiff behavior for the corresponding leg (as observed from the different colors representing the different stiff components), while the remaining activation weights stay close to zero.

The proposed approach does not only learn when to change the compliance in an on/off fashion, but also the manner to switch between the two behaviors. The sharp stiff/compliant difference is a characteristic of the collaborative task presented here (mostly binary, but with smooth transitions between the two compliance levels), which is correctly learned by the proposed approach. Fig. 7 shows the resulting stiffness matrix for the demonstration corresponding to leg 1, where the Cartesian robot position is shown along with the corresponding stiffness for each axis. We can see how these values vary along the different Cartesian axes, which is useful when the robot behavior demands to be stiff in a specific direction and compliant in the others.

In order to show the relevance of combining visual and haptic data for generalizing the learned task, two situations that did not appear in the demonstrations were presented to the robot (Fig. 8). First, $D_L$ tried to screw the leg at the center of the table, which means that the leg was placed at an incorrect position. In the second situation, $D_L$ positioned the leg in one of the table threads but the leg was tilted, making the screwing process unfeasible. In both cases, the robot behaved compliantly as expected, because neither corresponded to a correct screwing phase. A video of the experiment and the task-parameterized GMM sourcecode are available at http://programming-by-demonstration.org/AAAI2013/

5 Conclusions and Future Work

We presented a learning framework to encode and reproduce impedance behaviors using a task-parameterized statistical dynamical system. Our method allows to encode behaviors that rely on task variables, yielding only one model to encode the whole task. In contrast to previous approaches where robot impedance is learned from position variability, our framework extracts the impedance behavior from the manner in which the teacher behaves during the task, relying on recorded force patterns and visual information. We use forces not only to encode the skill, but also to estimate the stiffness of virtual springs governing the collaborative behavior, thus emphasizing that interaction forces-torques vary during different phases of the task.

The proposed approach is used to learn a reactive behavior, where the model automatically provides soft clusters of the different impedance behaviors that the robot might adopt. They are estimated as impedance controllers whose parameters are learned from demonstrations and associated with each model component. We plan in future work to provide the robot with a more active role. Initially, the roles would be considered as time independent. Then, more complex task preferences between the partners would be acquired, where the robot could adopt a proactive role in the collaboration. We devise to exploit the weighting mechanism in the model to influence the switching from reactive to proactive behaviors, by anticipating the next part of the task depending on user preferences. For example, if the user holding the leg is too far from the robot, it could take the initiatives to anticipate the movement and move to the leg.
References