Motion planning using first-order synergies

Néstor García, Jan Rosell and Raúl Suárez

Abstract— **This paper proposes a novel motion planning approach that exploits the concept of synergies (correlations) between degrees of freedom, extending it to the velocity space and calling them first-order synergies. An automatic partition method is defined to optimally divide the configuration space into cells where first-order synergies are significantly different. Using this partition, an algorithm that tends to grow a tree by extending the branches in the directions determined by the the first-order synergies of the cell where the leaf to be grown lies is introduced and called FOS-RRT. This allows the natural expansion of the tree along the directions determined by the data used to define the synergies. 2D examples illustrate the performance of the proposed approach, which is particularly attractive for potential applications in human-like robots using human synergies.**

I. INTRODUCTION

Motion planning is a traditional field in robotics [1], but, nevertheless, new problems are incessantly appearing due to continuous advances in the robot developments. New approaches are being proposed to solve them, as well as to improve the existing solutions to classical problems. A paradigmatic case is the humanoid robotics, since the advances done in this field require motion planners not only to look efficiently for an optimal solution in the classic way (optimizing energy or time in the plan execution) but also looking for human-like solutions, i.e. requiring the robot movement to be similar to those of the human being.

Successful approaches to motion planning for systems with many degrees of freedom (DOF) are those called *sampling-based planners* [2]. The most representative proposals being the *Probabilistic Road Map planners* (PRM) [3] and the *Rapidly-exploring Random Trees planners* (RRT) [4]. Following these pioneering proposals several variations were proposed improving or modifying some particular aspects; some examples are the consideration of optimality criteria [5] or task constraints [6], and the biasing of the sampling process towards regions of the sampling space that potentially could be more interesting [7], [8].

Trying to mimic the human movements while reducing the complexity of the search space for planning purposes lead to the concept of synergies, which could be conceptually defined as correlations between the DOF of the system. These correlations were used in the field of robotic hands. The idea was to look for the existing correlations in the human hand joints while grasping an object [9] and then try to replicate them with robotic hands [10] in order to reduce the dimension of the planning space, and therefore the complexity of the problem. This concept was recently applied to manipulation tasks using a two-arm anthropomorphic system [11]. Hand synergies were also considered with other related goals [12], and some relevant examples are the optimal identification of the hand pose using low-cost gloves [13], the design of the gloves for this purpose [14], the analysis and design of robotic hands in order to mimic human grasps [15], the selection of grasping forces [16], or the design of specific hand control systems [17][18]. Most of the mentioned works use synergies for grasp synthesis, thus the synergies are determined from the human hand analysis while performing graspings. Other approaches determine the synergies from freely movements of the human hand trying to cover the whole hand workspace without any external constraint [19], and then use the synergies for motion planning while mimicking human hand poses [20]. In addition, a related alternative work fixes artificial synergies to impose a common behavior to a team of mobile robots [21].

In this work we propose a new approach to motion planning exploiting the concept of synergies (as correlations among the system DOF), extending it to the velocity space. The final goal of the approach is to allow the system to learn the synergies of the human actions and use them for motion planning. As explained above, in previous works synergies were computed considering samples of the system to be replicated (typically the human being) captured in the configuration space. The approach we propose here complements the information embedded in these traditional synergies with information obtained from the computation of new synergies from samples captured in the velocity space of the system, which we call *first-order synergies* in comparison with those obtained from the configuration space, that can be considered as *zero-order synergies*. The paper presents the first-order synergies and demonstrative examples of their application in motion planning problems. The obtained results are satisfactory and encourage the application to different robotic problems, specially in the field of motion planning for humanoid robots.

After this introduction, Section II presents the concepts of zero- and first-order synergies, Section III presents the used nomenclature and an overview of the proposal, which is detailed in Sections IV and V. The approach is illustrated in Section VI and finally Section VII presents the conclusions and future work.

The authors are with the Institute of Industrial and Control Engineering (IOC), Universitat Politècnica de Catalunya (UPC), Barcelona, Spain, raul.suarez@upc.edu. This work was partially supported by the Spanish Government through the projects DPI2011-22471, DPI2013-40882-P and DPI2014-57757-R.

II. JOINT SYNERGIES

This section presents the concepts of zero- and first-order synergies, including a brief description of its computation.

A. Zero-order synergies

Zero-order synergies have already been given different names in the related bibliography, like "postural synergies", "eigengrasps", or "principal motion directions"; in this work we will use zero-order synergies as a generalization in contrast to the first-order synergies introduced in the next subsection.

Zero-order synergies represent correlations between DOF of the system under study (for instance the joint positions of a human hand or arms), and they are obtained from an analysis of a set of configuration samples of such system. The analysis of these samples is done with a *Principal Component Analysis* (PCA) that returns a new basis of the configuration space (eigenvectors) with the axes ordered according the dispersion of samples along it (eigenvalues). Each axis of this basis represents a zero-order synergy, i.e. the movement along one particular axis, equivalent to one single DOF, implies a correlated movement of several (or all) the actual DOF of the system. A basis of the configuration space representing zero-order synergies will be called a zero-order basis ${}^0S$.

B. First-order synergies

As a generalization of the zero-order synergies we introduce here the concept of first-order synergies, considering as such the result of applying the PCA to a set of *velocity samples* instead of configuration samples. We name these synergies as first order because they are obtained in the space of the first derivative of the configuration trajectories. i.e the velocity space. In practice, we sample the demonstration movements (for instance, of a human hand or arms) capturing the configurations with a given sampling rate, and approximate the velocity with the central finite difference method of second-order accuracy [22]. A PCA is done on this set of velocity samples, giving as a result a new basis of the velocity space, which in accordance with the previous reasonings, we call first-order basis ${}^1S$.

III. OVERVIEW OF THE PLANNING PROCEDURE

Let C be an ordered data set of configurations sampled from executions performed by an operator, and V the a velocity data set computed from C. Then, the planning procedure is as follows:

1) Scale the configurations in C so that each coordinate be in the range $[0, 1]$ and then apply the PCA to obtain the zero-order synergies ${}^0S$.
2) Scale the samples in V so that each coordinate be in the range $[-1, 1]$ and then apply the PCA to obtain the first-order synergies ${}^1S$.
3) Obtain a kd-tree partition of the relevant region of the configuration space determined by ${}^0S$, by consecutively dividing the cells whenever the first-order synergies of the two resulting subcells differ from each other (Section IV). A measure of the likeness between first-order synergies is defined for this purpose.
4) Given a planning query ($\boldsymbol{q}_{\text{init}}$, $\boldsymbol{q}_{\text{goal}}$) find a solution path using the FOS-RRT algorithm (Section V) that grows a tree taking into account the first-order synergies, i.e. the growing direction depends on the velocities defined by the first-order synergies of the cell where the leaf to be grown lies.

The following nomenclature will be used:

- m: Dimension of the configuration space.
- $U = [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]$: Matrix of eigenvectors resulting from a PCA (ordered by decreasing order of the corresponding eigenvalues).
- $\boldsymbol{\sigma} = [\sigma_1, \ldots, \sigma_m]$: Vector of eigenvalues (in decreasing order) resulting from a PCA.
- D: Diagonal matrix with the values of $\boldsymbol{\sigma}$ in the diagonal.
- $\boldsymbol{\mu}$: Barycenter resulting from the PCA.
- $\mathcal{N}$: Multivariate normal distribution of the samples, centered at $\boldsymbol{\mu}$ and with covariance matrix $\Sigma = UD^2U^\intercal$:

$$\mathcal{N}(\boldsymbol{\mu}, \Sigma) = (2\pi)^{-\frac{m}{2}} |\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(\boldsymbol{x}-\boldsymbol{\mu})^\intercal \Sigma^{-1}(\boldsymbol{x}-\boldsymbol{\mu})} \quad (1)$$

- S: Synergy basis $(\boldsymbol{\mu}, [\sigma_1\boldsymbol{u}_1, \ldots, \sigma_m\boldsymbol{u}_m])$.
- $B(S)$: m-dimensional box enclosing the 95% of the sample normal distribution, centered at $\boldsymbol{\mu}$ and with each side aligned with $\boldsymbol{u}_j$ and measuring $2\,\lambda\,\sigma_j$ with $\lambda = \sqrt{2}\ \text{erf}^{\,-1}(0.95^{\frac{1}{m}})$.
- $\mathcal{V}$: Volume of a box $B(S)$.
- $\mathcal{L}_{AB}$: Likeness measure between bases ${}^1S_A$ and ${}^1S_B$.

IV. CONFIGURATION SPACE PARTITION

A. Likeness of first-order bases

This subsection proposes a measure of the likeness $\mathcal{L}_{AB}$ of two first-order bases ${}^1S_A$ and ${}^1S_B$ using the multivariate normal distributions that each basis represents. $\mathcal{L}_{AB}$ is defined as a combination of the likeness $\mathcal{L}_{\boldsymbol{\mu}_{AB}}$ and $\mathcal{L}_{\Sigma_{AB}}$, all in the range $[0, 1]$, with 1 meaning maximum likeness:

$$\mathcal{L}_{AB} = \rho\,\mathcal{L}_{\boldsymbol{\mu}_{AB}} + (1-\rho)\,\mathcal{L}_{\Sigma_{AB}} \quad \text{with} \quad \rho \in [0,\, 1] \quad (2)$$

$\mathcal{L}_{\boldsymbol{\mu}_{AB}}$ measures the likeness of the barycenters $\boldsymbol{\mu}_A$ and $\boldsymbol{\mu}_B$, and $\mathcal{L}_{\Sigma_{AB}}$ measures the likeness of the directions of two synergy bases ${}^1S_A$ and ${}^1S_B$ and the size of their boxes $B({}^1S_A)$ and $B({}^1S_B)$. The value of ρ determines the behavior of $\mathcal{L}_{AB}$. In this work the weight $\rho = 0.2$ has been chosen empirically considering the differences between the barycenters ($\mathcal{L}_{\boldsymbol{\mu}_{AB}}$) as less important.

$\mathcal{L}_{\boldsymbol{\mu}_{AB}}$ evaluates both the direction and the magnitude similarities between barycenters (using respectively the variables $d_{\boldsymbol{\mu}_{AB}}$ and $m_{\boldsymbol{\mu}_{AB}}$ detailed below) and it is computed as:

$$\mathcal{L}_{\boldsymbol{\mu}_{AB}} = 1 - d_{\boldsymbol{\mu}_{AB}} m_{\boldsymbol{\mu}_{AB}} \in [0,\, 1] \quad (3)$$

with

$$d_{\boldsymbol{\mu}_{AB}} = \frac{1}{3}\left(2 - \frac{\boldsymbol{\mu}_A \cdot \boldsymbol{\mu}_B}{\|\boldsymbol{\mu}_A\|\|\boldsymbol{\mu}_B\| + \varepsilon}\right) \in \left[\frac{1}{3},\, 1\right] \quad (4)$$

$$m_{\boldsymbol{\mu}_{AB}} = 0.5\, m^{-\frac{1}{2}} \|\boldsymbol{\mu}_A - \boldsymbol{\mu}_B\| \in [0,\, 1] \quad (5)$$

where ε is the machine epsilon.

$\mathcal{L}_{\Sigma_{AB}}$ is computed as:

$$\mathcal{L}_{\Sigma_{AB}} = \frac{\Phi_{AB} - \Phi_{AB_{min}}}{\Phi_{AB_{max}} - \Phi_{AB_{min}}} \in [0,\, 1] \tag{6}$$

where Φ_{AB} is the value of the integral of the product of the two multivariate normal distributions $\mathcal{N}_A = \mathcal{N}(0,\, \Sigma_A)$ and $\mathcal{N}_B = \mathcal{N}(0,\, \Sigma_B)$.

Then, Φ_{AB} can be expressed as:

$$\Phi_{AB} = \int_{-\infty}^{\infty} \cdots \int_{-\infty}^{\infty} \mathcal{N}_A \mathcal{N}_B \, d\boldsymbol{x} = \left((2\pi)^{\frac{m}{2}} |\Sigma_A + \Sigma_B|^{\frac{1}{2}} + \varepsilon \right)^{-1} \tag{7}$$

and has the following bounds:

$$\Phi_{AB_{\text{min}}} \leq \Phi_{AB} \leq \Phi_{AB_{\text{max}}} \tag{8}$$

$$\Phi_{AB_{\text{min}}} = \left((2\pi)^{\frac{m}{2}} \prod_{j=1}^{m} \left(\sigma_{A_j} + \sigma_{B_{m-j+1}} \right) + \varepsilon \right)^{-1} \tag{9}$$

$$\Phi_{AB_{\text{max}}} = \left(\pi^{\frac{m}{2}} \prod_{j=1}^{m} \left(\sigma_{A_j} + \sigma_{B_j} \right) + \varepsilon \right)^{-1} \tag{10}$$

with σ_{A_j} and σ_{B_j} being the eigenvalues of the synergies ${}^1S_A$ and ${}^1S_B$ respectively.

B. Partition criteria

The box of the global zero-order basis $B({}^0S)$ is split into cells of a kd-tree by hyperplanes aligned with the axes of ${}^0S$. The kd-tree is built by recursively dividing a cell (called parent cell) into two subcells (called left and right child cells). The partition criterion is based on the differences between the first-order basis of the parent cell ${}^1S_P$ and the first-order bases of its left and right child cells ${}^1S_L$ and ${}^1S_R$ respectively. The position x of the partition hyperplane along the partition axis is the one that optimizes the following two objective functions:

1) Minimization of the maximum likeness between the parent basis and those of the children:

$$O_{\mathcal{L}} = \max\left(\mathcal{L}_{PL},\, \mathcal{L}_{PR}\right) \in [0,\, 1] \tag{11}$$

This objective function pursues that both child first-order synergies be different to the parent synergies.

2) Minimization in the velocity space of the maximum of the volumes ${}^1\mathcal{V}_L$ and ${}^1\mathcal{V}_R$ of the child boxes $B({}^1S_L)$ and $B({}^1S_R)$, respectively, normalized to the volume ${}^1\mathcal{V}_P$ of the parent box $B({}^1S_P)$:

$$O_{\mathcal{V}} = \frac{\max\left({}^1\mathcal{V}_L,\, {}^1\mathcal{V}_R\right)}{{}^1\mathcal{V}_P} \in [0,\, 1] \tag{12}$$

This objective function aims at enclosing the velocity samples of the parent cell within the boxes of the child cells tighter than within the parent box.

Note that both $O_{\mathcal{L}}$ and $O_{\mathcal{V}}$ lie in the range $[0,\, 1]$ and that $O_{\mathcal{L}}$ and $O_{\mathcal{V}}$ are equal to 1 in the limits of the x-domain (when one of the children is equal to the parent and the other is void). Therefore, there exists always a minimum (unless the objective function were completely flat).

These two objective functions are joined to get a general objective function as follows:

$$O = \frac{O_{\mathcal{L}}}{th_{\mathcal{L}}} + \frac{O_{\mathcal{V}}}{th_{\mathcal{V}}} \tag{13}$$

Algorithm 1: KD-TREE

Input : Samples M
Thresholds $th_{\mathcal{L}}$, $th_{\mathcal{V}}$
Output: Kd-tree $\mathcal{T}$

for $j \leftarrow 1$ **to** m **do**
  $\pi_j \leftarrow$ MINIMIZEO$(M,\, j,\, th_{\mathcal{L}},\, th_{\mathcal{V}})$
SORT$(\boldsymbol{\pi})$
for $k \leftarrow 1$ **to** m **do**
  if VALIDPARTITION(π_k) **then**
    $(th_{\mathcal{L}}, th_{\mathcal{V}}) \leftarrow$ UPDATETH$(th_{\mathcal{L}}, th_{\mathcal{V}}, \boldsymbol{\pi})$
    $\mathcal{T}.L \leftarrow$ KD-TREE(LEFTS$(M, \pi_k), th_{\mathcal{L}}, th_{\mathcal{V}})$
    $\mathcal{T}.R \leftarrow$ KD-TREE(RIGHTS$(M, \pi_k), th_{\mathcal{L}}, th_{\mathcal{V}})$
    return $\mathcal{T}$
return $\mathcal{T}$

where $th_{\mathcal{L}}$ and $th_{\mathcal{V}}$ weight the combination of the two objective functions and are threshold values used to accept a partition: the partition is accepted as valid only when the position x of the partition hyperplane that minimizes O satisfies $O_{\mathcal{L}} < th_{\mathcal{L}}$ and $O_{\mathcal{V}} < th_{\mathcal{V}}$; otherwise the parent cell is not further divided.

The search of the optimum partition position is repeated for all the axis and the selected splitting hyperplane is the one with lowest O value among the valid partitions.

C. Partition algorithm

The partition procedure, called KD-TREE, is detailed in Algorithm 1 and uses the functions detailed below. Let π_j be a partition hyperplane (perpendicular to the j-axis) that includes information of the associated values O, $O_{\mathcal{L}}$ and $O_{\mathcal{V}}$ of the optimization functions corresponding to this partition, and let $\boldsymbol{\pi}$ be a m-dimensional vector $(\pi_1, \ldots, \pi_m)$.

- MINIMIZEO$(M,\, j,\, th_{\mathcal{L}},\, th_{\mathcal{V}})$: Locates the minimum of O using Brent's algorithm [23] and returns the hyperplane π_j at the position that minimizes O. If a minimum is not found, the hyperplane is set at an arbitrary position and the associated O value is set at a very high arbitrary value.
- SORT$(\boldsymbol{\pi})$: Sorts the elements of $\boldsymbol{\pi}$ by ascending order of the associated optimization value O.
- VALIDPARTITION(π): Returns true if the resulting child cells are wide enough and both $O_{\mathcal{L}}$ and $O_{\mathcal{V}}$ are below their respective thresholds.
- UPDATETH$(th_{\mathcal{L}}, th_{\mathcal{V}}, \boldsymbol{\pi})$: Takes $O_{\mathcal{L}max}$, the largest value of $O_{\mathcal{L}}$ among the ones obtained when minimizing O along each axis, and if $th_{\mathcal{L}} > O_{\mathcal{L}max}$, sets $th_{\mathcal{L}} \leftarrow O_{\mathcal{L}max}$. The same is done for $th_{\mathcal{V}}$.
- LEFTS$(M,\, \pi)$ and RIGHTS$(M,\, \pi)$: Return the samples of the set M which lie in the positive and negative side of the hyperplane π, respectively.

The main features of the KD-TREE algorithm are:

▷ The parameters $th_{\mathcal{L}}$ and $th_{\mathcal{V}}$, the thresholds for the objective functions $O_{\mathcal{L}}$ and $O_{\mathcal{V}}$ respectively, are used to accept or not a partition. The first call to the algorithm uses $th_{\mathcal{L}}$ and $th_{\mathcal{V}}$ set to 1. Then, in the next calls,

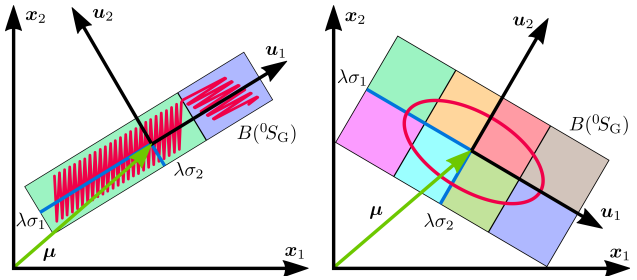

Fig. 1. A ray-shaped motion (left) and an elliptic motion (right) and the resulting kd-trees built by splitting the corresponding boxes $B(^0S)$ into two and eight cells, respectively, with hyperplanes aligned with the ^0S axes.

when a cell is divided, these values are updated (Line 7) before calling recursively the algorithm with the samples of the left and right child cells. Since the threshold values can only decrease, the acceptance criteria for new partitions becomes stricter as the kd-tree grows.

- ▷ M contains all the samples that lie in the cell. Every sample is a pair with a position and a velocity vectors.
- ▷ (Lines 6 to 11) If a valid partition is found, the cell is divided and the left and right subtrees are assigned to the child cells L and R, respectively.

D. Examples

Fig. 1 (left) shows the global zero-order basis ^0S of the samples of a motion that follows a ray shape with two parts where the motion directions differ from each other. The figure shows the result of the KD-TREE algorithm that partitions ^0S into two cells. Fig. 1 (right) shows the global zero-order basis ^0S of the samples of a motion that follows an elliptic trajectory counter-clock wise. In this case the partition algorithm found a kd-tree composed of 8 cells.

V. PLANNING PROCEDURE

A. The FOS-RRT algorithm

The basic RRT planning procedure [4] builds a tree $\mathcal{T}$ rooted at an initial configuration $\boldsymbol{q}_{\text{init}}$, by consecutively sampling a configuration $\boldsymbol{q}_{\text{rand}}$, selecting the nearest node $\boldsymbol{q}_{\text{near}}$ in $\mathcal{T}$ and growing a rectilinear segment of maximum length ϵ towards $\boldsymbol{q}_{\text{rand}}$ (the function that grows the tree is usually called EXTEND). With a small probability, the sampling function returns $\boldsymbol{q}_{\text{goal}}$ in order to bias the growing towards the goal and thus accelerate the search of the solution path. Once the tree reaches the goal configuration the solution path is retrieved by backtracking the parent relationships in $\mathcal{T}$ from $\boldsymbol{q}_{\text{goal}}$ to $\boldsymbol{q}_{\text{init}}$.

In this work the basic RRT planning procedure has been modified by changing the EXTEND function. The proposal takes into account the first-order synergies in the region where the node $\boldsymbol{q}_{\text{near}}$ lies in order to modify the growing direction. The new planning procedure, called FOS-RRT or *First-order Synergy*-RRT, is shown in Algorithm 2. It can be seen that the tree is grown using the FOS-EXTEND function, described in Algorithm 3. The functions used by these algorithms are:

Algorithm 2: FOS-RRT

```
Input  : Configurations q_init, q_goal
         Probability P_goal
Output: Path
1: T.Init(q_init)
2: for j = 1 to K do
3:     q_rand ← RANDOMCONFIG(q_goal, P_goal)
4:     if FOS-EXTEND(T, q_rand, q_goal) then
5:         return PATH(T, q_init, q_goal)
6: return ∅
```

- RANDOMCONFIG($\boldsymbol{q}_{\text{goal}}$, P_{goal}): Returns with probability $1 - P_{\text{goal}}$ a random configuration of the configuration space, and with probability P_{goal} returns $\boldsymbol{q}_{\text{goal}}$.
- PATH($\mathcal{T}$, $\boldsymbol{q}_{\text{init}}$, $\boldsymbol{q}_{\text{goal}}$): Returns the solution path as a sequence of configurations obtained by backtracking the parent relationships in the nodes of $\mathcal{T}$ from $\boldsymbol{q}_{\text{goal}}$ to $\boldsymbol{q}_{\text{init}}$.
- NEARESTNEIGHBOR($\boldsymbol{q}$,$\mathcal{T}$): Returns the node of $\mathcal{T}$ closest to $\boldsymbol{q}$.
- FOS-BASIS($\boldsymbol{q}$): Returns the first-order basis of the cell where the configuration $\boldsymbol{q}$ lies, or $\emptyset$ if it lies outside $B(^0S)$.
- SCALE($\boldsymbol{w}$): Returns a velocity with all the components in the range $[-1, 1]$ which is obtained from $\boldsymbol{w}$ in two steps. First, each coordinate of $\boldsymbol{w}$ is divided by the maximum velocity in the corresponding axis to obtain $\boldsymbol{w}_{\text{scale}}$. Second, while there exists a $\boldsymbol{w}_{\text{scale},j}$ with $|\boldsymbol{w}_{\text{scale},j}| > 1$, $\boldsymbol{w}_{\text{scale}}$ is divided by $|\boldsymbol{w}_{\text{scale},j}|$.
- REAL($\boldsymbol{v}$): Returns the velocity resulting from multiplying each component of $\boldsymbol{v}$ (that lies in the range $[-1, 1]$) by the maximum velocity in the corresponding axis.
- RANDTRIA(k_1, k_2, k_3): Returns a random value of a triangular distribution with minimum at k_1, mode at k_2 and maximum at k_3.
- COLLISIONFREE($\boldsymbol{q}_1, \boldsymbol{q}_2$): Returns true if the rectilinear segment between $\boldsymbol{q}_1$ and $\boldsymbol{q}_2$ is collision-free, and false otherwise.

The main features of the FOS-EXTEND function are:

- ▷ The ϵ parameter defines the standard advance step of the standard RRT. The Δt parameter defines the time step used to advance with the velocity defined by the synergies. An empirical value of Δt that gives good results is $\Delta t = 20 \frac{\epsilon}{\|\boldsymbol{v}_{\max}\|}$ where $\boldsymbol{v}_{\max}$ is the velocity when all the DOF are actuated at their maximum speed.
- ▷ (Line 4-5) FOS-EXTEND behaves as the basic EXTEND function (the tree grows like the standard RRT) when:
  a) $\boldsymbol{q}_{\text{near}}$ is outside the box $B(^0S)$ (since in this case no first-order basis is available).
  b) $\boldsymbol{q}_{\text{rand}}$ is equal to $\boldsymbol{q}_{\text{goal}}$ (in order to bias the growing and to guarantee the goal reaching).
  c) $\boldsymbol{v}$ points in the opposite direction to $\boldsymbol{\mu}$ (the growth along non-natural directions should be restrained).
- ▷ (Line 7) $\boldsymbol{v}_\epsilon$ is a velocity defined by $\boldsymbol{q}_{\text{rand}}$ that allows to advance a distance ϵ when it is applied during Δt.

Algorithm 3: FOS-EXTEND

Input : RRT $\mathcal{T}$
Configurations $\boldsymbol{q}_{\text{rand}}$, $\boldsymbol{q}_{\text{goal}}$
Output: Bool

1: $\boldsymbol{q}_{\text{near}} \leftarrow$ NEARESTNEIGHBOR($\boldsymbol{q}_{\text{rand}}$, $\mathcal{T}$)
2: ${}^1S_{\text{near}}(\boldsymbol{\mu}, \boldsymbol{\sigma}, [\boldsymbol{u}_1, \ldots, \boldsymbol{u}_m]) \leftarrow$ FOS-BASIS($\boldsymbol{q}_{\text{near}}$)
3: $\boldsymbol{v} \leftarrow$ SCALE $\left(\frac{\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}}{\Delta t}\right)$
4: **if** ${}^1S_{near} = \emptyset$ **or** $\boldsymbol{q}_{rand} = \boldsymbol{q}_{goal}$ **or** $\boldsymbol{\mu} \cdot \boldsymbol{v} < 0$ **then**
5: $\quad \boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{near}} + \min(\epsilon, \|\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}\|) \frac{\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}}{\|\boldsymbol{q}_{\text{rand}} - \boldsymbol{q}_{\text{near}}\|}$
6: **else**
7: $\quad \boldsymbol{v}_\epsilon \leftarrow \frac{\boldsymbol{v}}{\|\boldsymbol{v}\|} \frac{\epsilon}{\Delta t}$
8: $\quad \boldsymbol{v}_{\text{FOS}} \leftarrow \boldsymbol{\mu} + \sum_{j=1}^{m} \boldsymbol{u}_j \, \sigma_j \left(\frac{\boldsymbol{v} - \boldsymbol{\mu}}{\|\boldsymbol{v} - \boldsymbol{\mu}\|} \cdot \boldsymbol{u}_j\right)$
9: $\quad c \leftarrow$ RANDTRIA(0, 1, 1)
10: $\quad \boldsymbol{v}_{\text{c}} \leftarrow (1 - c)\, \boldsymbol{v}_\epsilon + c\, \boldsymbol{v}_{\text{FOS}}$
11: $\quad \boldsymbol{q}_{\text{new}} \leftarrow \boldsymbol{q}_{\text{near}} + \Delta t$ REAL($\boldsymbol{v}_{\text{c}}$)
12: **if** COLLISIONFREE($\boldsymbol{q}_{near}$, $\boldsymbol{q}_{new}$) **then**
13: $\quad \mathcal{T}$.AddVertex($\boldsymbol{q}_{\text{new}}$)
14: $\quad \mathcal{T}$.AddEdge($\boldsymbol{q}_{\text{near}}$, $\boldsymbol{q}_{\text{new}}$)
15: $\quad$ **return** $\boldsymbol{q}_{\text{new}} = \boldsymbol{q}_{\text{goal}}$
16: **return false**

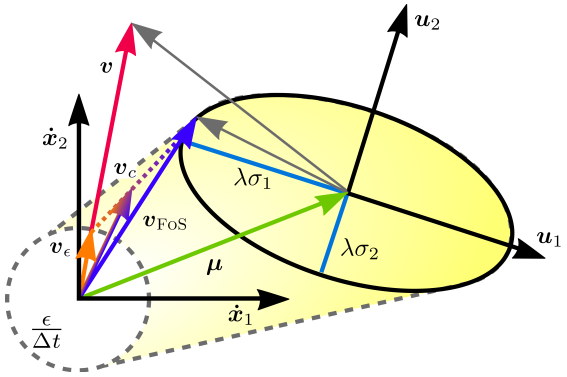

Fig. 2. Example of the region where $\boldsymbol{v}_c$ lies, given the first-order basis ${}^1S$.

- ▷ (Line 8) $\boldsymbol{v}_{\text{FOS}}$ is a velocity defined by the synergies. It is computed with the barycenter and a weighted projection (onto the synergy basis) of the $\boldsymbol{q}_{\text{rand}}$ direction (Line 3).
- ▷ (Line 11) $\boldsymbol{q}_{\text{new}}$ is computed from $\boldsymbol{q}_{\text{near}}$ by moving during Δt with the velocity $\boldsymbol{v}_c$ defined as a linear combination of $\boldsymbol{v}_{\text{FOS}}$ and $\boldsymbol{v}_\epsilon$ (Line 10). The parameter of the linear combination is extracted from a triangular distribution (Line 9) that gives more weight to $\boldsymbol{v}_{\text{FOS}}$.

As an example Fig. 2 shows in yellow the region where all the possible $\boldsymbol{v}_{\text{c}}$ lie. The figure also illustrates the first-order basis ${}^1S$ and the vectors $\boldsymbol{v}$, $\boldsymbol{v}_\epsilon$ and $\boldsymbol{v}_{\text{FOS}}$.

B. Planning examples

The proposal has been implemented within The Kautham Project [24], a motion planning and simulation environment developed at the Institute of Industrial and Control Engineering (IOC-UPC) for teaching and research.

Fig. 3 shows the results (the tree and the solution path) of two simple examples solved using the FOS-RRT algorithm. On Fig. 3 (left) two regions with different first-order bases have been artificially defined (the region on the left has its principal direction along the vertical axis while the region on the right has it along the horizontal one). The edges of the tree grow according to these main directions.

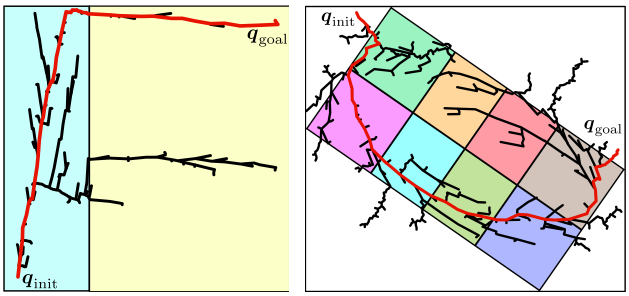

Fig. 3. Planning examples using FOS-RRT: it can be seen that the tree branches grow with a greater pace in the senses defined by the first-order synergies of each of the cells (in red the solution path).

On Fig. 3 (right) $B({}^0S)$ is divided into eight cells, each with its corresponding first-order basis. These bases make the edges of the tree to grow following an elliptic motion counter-clock wise. It can be noticed that the branches in the counter-flow sense or outside the box $B({}^0S)$ advance with a slower step size.

VI. APPLICATION EXAMPLE

To illustrate the proposed approach a 2D example has been designed, it consists of a maze. The problem to be solved is how to go from one point to another, and the goal of the proposed approach is to teach the system by showing the preferred movements of the operator. Experimental data of position has been recorded while moving a magnetic tracker repetitively along the corridors in a counterclockwise sense, as shown in Fig. 4 (left). The PCA has been run and both the zero- and first-order synergies have been obtained and used to run the partition procedure. The resulting partitioned configuration space is shown in Fig. 4.

Then, a query has been set with $\boldsymbol{q}_{\text{init}}$ at the top-right corner and $\boldsymbol{q}_{\text{goal}}$ at the bottom-right one, and run using the FOS-RRT, the standard RRT and the KPIECE [25] planners. The results obtained are shown in Fig. 4. It can be appreciated how the proposed approach builds a tree whose branches grow with a greater pace in the sense defined by the first-order synergies, thus encountering a solution more similar to those of the demonstration set than the one found using the standard RRT (mainly with fixed-length branches) or the KPIECE (with branches with a length that varies randomly within a given interval). For illustrative purpose, Table I shows the average results after 100 executions obtained for this example using the mentioned algorithms. The comparison parameters are the success rate, the number of tree nodes, the used time (when running in a 3.40-GHz Intel i7-3770, 8-GB RAM PC), the solution length, the percentage of times that the extend functions returned true and the number of collisions checks. Even when the FOS-RRT solution has a larger length it was found in a shorter time and follows better the human demonstrations.

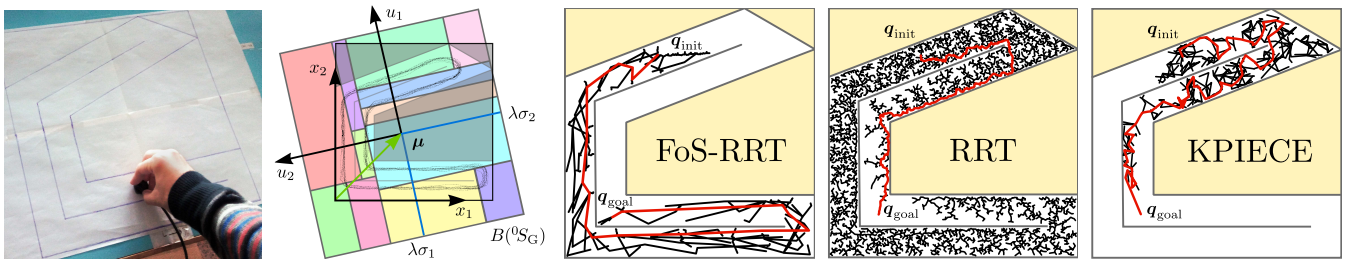

Fig. 4. From left to right: Data acquisition using a magnetic tracker; $B(^0S_G)$ partition; tree and solution path found using different planners.

TABLE I

AVERAGE RESULTS OF THE MOTION PLANNING.

Planner	Success rate	Tree nodes	Used time (s)	Solution length (m)	Valid segments	Collision checks
FoS-RRT	100%	2283	1.125	1.165	45.69%	16088
RRT	100%	7013	1.924	0.942	36.06%	23186
KPIECE	100%	1145	1.483	1.469	82.39%	5794

VII. CONCLUSIONS AND FUTURE WORK

This paper has introduced the concepts of zero- and first-order synergies describing the correlation between the configurations and velocities of the robot DOF, respectively. These synergies can be, for instance, computed from human operator demonstrations, or fixed before hand following some criteria. Since the first-order synergies may vary along the configuration space, an automatic partition procedure of the relevant part of the configuration space, determined by the zero-order synergies, has been proposed that optimizes obtaining cells with significantly different first-order synergies. A novel planning approach, that modifies the standard RRT growing, has been introduced. The proposal defines the growing direction taking into account the first-order synergies of the cell where the leaf to be grown lies, allowing in this way the natural expansion of the tree along the directions determined by the captured data. The results obtained show that the proposal works well and the procedure has been illustrated in a 2D example.

Future work is centered in applying the method to the motion planning for a dual-arm robotic system. The data of a human operator performing different tasks with both arms have to be captured and mapped to the robotic system, where the PCA to obtain the zero-order and the first-order synergies has to be computed. These synergies will capture the motion velocities in the different regions of the space and its use with the FoS-RRT planning algorithm is expected to give human-like solutions quite efficiently.

REFERENCES

[1] J. Latombe. *Robot motion planning*. Kluwer Academic Pub., 1991.

[2] Mohamed Elbanhawi and Mitar Simic. Sampling-based robot motion planning: A review. *Access, IEEE*, 2:56–77, 2014.

[3] L. E. Kavraki, P. Svestka, J. Latombe, and M. Overmars. Probabilistic Roadmaps for Path Planning in High-Dimensional Configuration Spaces. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 566–580, 1996.

[4] J. J. Kuffner and S. LaValle. RRT-Connect: An efficient approach to single-query path planning. In *Proc. of the IEEE Int. Conf. Robotics and Automation*, pages 995–1001, 2000.

[5] S. Karaman and E. Frazzoli. Sampling-based algorithms for optimal motion planning. *International Journal of Robotics Research*, 30(7):846–894, June 2011.

[6] M. Stilman. Global manipulation planning in robot joint space with task constraints. *IEEE Transactions on Robotics*, 26(3):576–584, 2010.

[7] A. Yershova, L. Jaille, T. Simeon, and S. LaValle. Dynamic-Domain-RRTs: Efficient Exploration by Controlling the Sampling Domain. In *Proc. IEEE Int. Conf. Robotics and Automation*, pages 3856–3861, 2005.

[8] J. Rosell, R. Suárez, and A. Pérez. Path planning for grasping operations using an adaptive PCA-based sampling method. *Autonomous Robots*, 35(1):27–36, 2013.

[9] M. Santello, M. Flanders, and J. F. Soechting. Postural hand synergies for tool use. *Journal of Neuroscience*, 18(23):10105–10115, 1998.

[10] M. T. Ciocarlie and P. K. Allen. Hand Posture Subspaces for Dexterous Robotic Grasping. *The International Journal of Robotics Research*, 28(7):851–867, July 2009.

[11] R. Suárez, J. Rosell, and N. García. Using synergies in dual-arm manipulation tasks. In *Proc. IEEE Int. Conf. Robotics and Automation*, page 5655, 2015.

[12] A. Bicchi, M. Gabiccini, and M. Santello. Modeling natural and artificial hands with synergies. *Philosophical Transactions of the Royal Society B*, 366:3153 – 3161, 2011.

[13] M. Bianchi, P. Salaris, and A. Bicchi. Synergy-based hand pose sensing: Reconstruction enhancement. *International Journal of Robotics Research*, 2012.

[14] M. Bianchi, P. Salaris, and A. Bicchi. Synergy-based hand pose sensing: Optimal glove design. *International Journal of Robotics Research*, 2012.

[15] F. Ficuciello, G. Palli, C. Melchiorri, and B. Siciliano. Postural synergies of the UB hand IV for human-like grasping. *Robotics and Autonomous Systems*, 62(4):515–527, 2014.

[16] M. Gabiccini, A. Bicchi, D. Prattichizzo, and M. Malvezzi. On the role of hand synergies in the optimal choice of grasping forces. *Autonomous Robots*, 31:235 – 252, 2011.

[17] T. Wimböck, B. Jan, and G. Hirzinger. Synergy-level impedance control for a multifingered hand. In *Proc. IEEE Int. Conf. on Intelligent Robots and Systems*, pages 973–979, 2011.

[18] G. Palli, C. Melchiorri, G. Vassura, U. Scarcia, L. Moriello, G. Berselli, A. Cavallo, G. De Maria, C. Natale, S. Pirozzi, C. May, F. Ficuciello, and B. Siciliano. Dexmart hand: Mechatronic design and experimental evaluation of synergy-based control for human-like grasping. *The International Journal of Robotics Research*, 33(5):799–824, 2014.

[19] S. Sun, C. Rosales, and R. Suárez. Study of coordinated motions of the human hand for robotic applications. In *Proc. of the IEEE Int. Conf. on Information and Automation*, pages 776–781, 2010.

[20] J. Rosell, R. Suárez, C. Rosales, and A. Pérez. Autonomous motion planning of a hand-arm robotic system based on captured human-like hand postures. *Autonomous Robots*, 31(1):87–102, 2011.

[21] J. Rosell and R. Suárez. cRRT*: Planning loosely-coupled motions for multiple mobile robots. In *19th IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*.

[22] B. Fornberg. *Generation of Finite Difference Formulas on Arbitrarily Spaced Grids*, volume 51, pages 699–706. 1988.

[23] R. P. Brent. *Algorithms for Minimization without Derivatives*, chapter 4. Prentice-Hall, Englewood Cliffs, NJ, 1973.

[24] J. Rosell, A. Pérez, A. Aliakbar, Muhayyuddin, L. Palomo, and N. García. The Kautham Project: A teaching and research tool for robot motion planning. In *IEEE Int. Conf. on Emerging Technologies and Factory Automation, ETFA'14*, 2014.

[25] Ioan A. Şucan and Lydia E. Kavraki. Kinodynamic motion planning by interior-exterior cell exploration. In *Algorithmic Foundation of Robotics VIII*, pages 449–464. Springer, 2010.