PERATECH

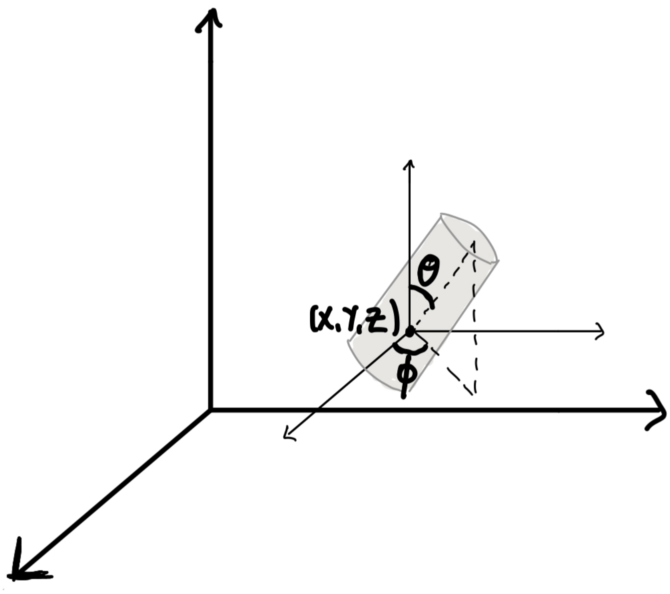
Efficient conversion from spatial coordinate to network connection representation

Dr Tim Wiles, Peratech Holdco Limited

The problem

An aspect of Peratech's sensing technology relies upon understanding connectivity properties of collections of cylindrical objects placed in a box.

Set-up A configuration of a finite number of identical, uniform density cylinders are placed in a box $B \subset \mathbf{R}^3$ of arbitrary length, width, and depth. The cylinders cannot overlap, and have identical diameter D and length L. The centre of mass (X, Y, Z) and rotation (θ, ϕ) of each cylinder is known. Rotations around the cylinder axes are irrelevant to this problem, due to symmetry, leading to only the five degrees of freedom above.



Two cylinders are said to have a *connection* if the minimum distance between them is smaller than a cut-off length, l_c. If a connection exists, we write d_{min} for the minimum distance between the objects. This also defines two *nodes*, one on each object, which are the points achieving this minimum distance.[1].

[1]Suppose that the configuration is such that these nodes are uniquely defined

Note that this gives rise to a finite collection of nodes, resulting from all connected pairs of cylinders in the box. We use this to build a *network* generated by the configuration.

First, we describe the inter-cylinder connectivity. Any pair of nodes in the network arising from the procedure above (i.e., from two cylinders with separation distance $d_{\min} \leq l_c$) are connected by an edge. We also associate a *weight* to this edge of $C(d_{\min})$, where C is an explicit function.

While $C(d_{\min})$ describes one major factor determining connectivity in the network, the remaining factor is the length between nodes in the cylinders. We therefore also join any two nodes corresponding to points *on the same cylinder* by an edge. The weight of such an edge is given by a function $P(d_{\min})$ of the Euclidean distance $d_{\min}$ between the the associated points.

Question Given a list of (X, Y, Z, θ, ϕ) for each cylinder in the configuration, what is the most efficient way to produce the network above. That is, to give a complete description of the associated nodes, edges and weights.

A Python script demonstrating this process would be an ideal way to convey successful solution of this problem.

Extensions

- What happens if the diameter D and length L is allowed to vary between cylinders?
- An quantity of interest is the Cheeger constant, or conductance, of the resulting weighted network (i.e., graph). Is there a more efficient approach to calculating or estimating this directly from the configuration of cylinders?
- In the question above, the configuration of cylinders is given. In fact, these configurations are currently generated via simulation, starting from an initial configuration of cylinders with deterministic positions but random orientations, and then running an interaction dynamic. This produces a random final configuration of cylinders exhibiting various desired properties. Is there a more efficient way to produce a configuration with these desired properties, for example, using Markov chains? (More details on the properties can be provided).