Double derivative controller

Idea: Let’s start from the same assumptions of the previous controller:

1. We want to find an invertible map between input and derivatives of the output
2. Linearize the system using this map

It is sufficient to consider second order derivatives of the output to find a non-singular map with the input. Let’s look at the map on the slide and we can evince that the matrix is non-singular for v different from 0. That is the only constraint that we have been imposed by theoretical analysis.

Now, in order to exploit the benefit of this input transformation, we consider the following change of state. This allows us to derive through error dynamics a linear control structure, in particular PD action, with feedforward.

Before moving on we point out two characteristics of the controller

* We have decoupled dynamics since u1 = z\_3dot and u2 = z\_4dot
* We have a practical constraint on the desired trajectory that must be smooth and persistent. Smoothness is required to avoid troubles with double derivatives and persistency to avoid singularity issues.

Let’s now have a look at the simulations results:

* The system is properly tracking the final trajectory 94% of the times
* Sensitivity proved to be an issue of this controller which lost 16% convergence score
* Proximity test is reasonable if compared with other controllers
* As we can see in the picture the controller provides a really smooth path for the agent. We can’t see changes of directions or backup manuevers

Being this controller one of the most complex in terms of implementations we will reserve the next two slides discussing the implementation issues:

* As said before derivatives in real systems cannot be implemented with continuous time block. Plus continuous time derivative causes instability issues in the Simulink simulations. So we have to implement real derivatives through a first order high pass filter. Pole allocation of the filter was arbitrary, but we point out that its choice is crucial for the final action of the controller. Changing the filter pole requires a tuning of the controller from scratch.   
  From empirical analysis we observed that most of the issues related to continuous time derivatives were given by differentiating the system output and not the desired trajectory. This is not surprising since the desired trajectory is circular (smooth and persistent).
* After that we had to address the previously discussed constraint of linear velocity, which must be different from zero to avoid singularities of the input transformation matrix. The solution was simply implemented by setting the minimum value of the linear velocity to 1e-4.

CONCLUSION

Well, now we are the ending of our presentation and we briefly recap our work. Using consensus we defined a randez-vous point for the agents and through regulation we allowed the agents to get closer to average consensus. Finally switching to a tracking control we made the agents follow a circular trajectory of radius 1. Tracking control proved to be the most challenging; hence we are now revising the main properties and results of our 4 controllers.

Once again we recall that the conclusions on the slides rely on out test suite and they are not theoretical results, but they represent a good guideline for practical implementation. Before we discussed about sensitivity, convergence and smoothness; what about convergence time? Let’s look at the data and we can easily evince that non-linear controller and sagittal one stand out.

Finally we present a table with the main properties of every controller. Its clear from these two tables that we cannot clearly decide which is the best controller. Every controller present its pros and cons. We would like to point out just to final conclusions:

* Non-linear controller proved to be the right choice if we look for robustness, convergence time and ease of implementation.
* Sagittal controller was the best overall if we consider the results of our test suite and it is the right choice if we look for a fast, easy tuning and smooth controller.