iea wind Task 52

LAC Summer Games 2024

Authors: David Schlipf, Feng Guo, Zhi Liang, Mark Pitter, Javier Lopez Queija, Jenni Rinker, Aravind Venkatachalapathy, Simon Weich

Content

Introduction

Wind characteristics are inherently dynamic, subject to sudden changes in speed, direction, and turbulence. Consequently, wind turbines are highly dynamic systems, excited by such stochastic influences and must be designed to handle these variations and disturbances. Traditional controllers react only after wind changes have impacted the wind turbine, resulting in an offset operation until the controller handles the disturbance. Contrary, lidar-assisted control (LAC) concepts use preview information of the wind to anticipate changes, hence reacting before the disturbance approaches the rotor and therefore improving the operation of the turbine.

The mission of our IEA Wind Task 52 working group is to push forward the lidar-assisted control technology and simplify its application by recommended practices, open-source tools and joined exercises. In line with our mission, we are very glad to launch our LAC Summer Games 2024 to encourage students and professionals in the art of designing and deploying lidar data processing algorithms and lidar-assisted controllers.

By participating in the games, you can increase your knowledge in LAC and enjoy the design process by competing with others. Moreover, we aim to trigger your creativity in three different disciplines and with it motivate the development of new concepts. During the competition, it is intended that the participants learn the existing open-source tools and their application will result eventually in the further development of the tools. Additionally, during the games you will be able to exchange knowledge with different parties of the wind energy community. We are looking forward to enhancing your passion for enhancing the vision of modern wind turbines and getting the best LAC out of you!

The tools are based on previous publications [1], [2] and built up on the IEA 15 MW reference wind turbine [3] and the open-source controller ROSCO [4]. We intend to write a publication on the results and integrate the learnings in our planned IEA Recommended Practices on Lidar-Assisted Control.

Our Categories

For the Summer Games we have the following categories:

1. Students (including PhD Students)
2. Researchers

Participation in groups is also possible. If all group members are students, the first category applies.

Evaluation and awards

In every discipline (see below) we will rank the contributions for each of the two categories mentioned above depending on the cost: for the 30 s sprint and the 18 m/s Hurdles, the lowest cost will win (minimization problem) for the DLC 1.2 Marathon, the highest cost will win (maximization problem). The top three candidates in each category and discipline will be asked to submit a DLL, so that results can be reproduced. The winners in each category will be announced on the IEA Task LinkedIn page and will also have the opportunity to present their solution to the IEA Task 52 group, which consists of a selection of industry experts.

Our Disciplines

In this section, you will find the different disciplines where you can participate with your proposed lidar-data processing algorithms and lidar-assisted controllers. Regardless of your chosen category, we encourage you to submit outcomes for as many disciplines as you believe align with your capabilities. In the following subsections, each of the disciplines are presented together with the evaluation methods used for evaluating the performance of the algorithms you send.

The 30 s Sprint

In the first challenge, like a sprint for a 100-meter, we'll test how well your controller handles an Extreme Operational Gust (EOG) at 12 m/s. The challenge is addressing mainly control experts and the intention is to introduce you to LAC and to trigger new control concepts. In this discipline, your controller response will be tested under the most challenging conditions for any controller evaluation. The objective is to cancel out the effect of the gust and to have a smooth rotor speed and tower top motion. Here, the wind preview is provided by a simple single point lidar, resulting in a five second preview.

The script RunExample.m/.py in the Release/IEA15MW_01 folder provides an example using a baseline collective pitch feedforward control [1], which is designed to compensate the impact of the gust on the rotor speed and works well for high wind speeds but provides plenty of room for improvements for wind speeds close to rated wind, see Figure 1. To simplify the task, we reduced the degrees of freedom of the wind turbine to rotor motion (GenDOF) and first fore-aft tower bending-mode (TwFADOF1).

The evaluation criteria for this discipline is the minimization of the combined relative reduction of the rotor and tower motion defined as:

$$J = \frac{\max(|\Omega - \Omega_0|)}{\Omega_0} + \frac{\max(|\mathrm{M}_{y\mathrm{T}} - \mathrm{M}_{y\mathrm{T}0}|)}{\mathrm{M}_{y\mathrm{T}0}},$$

where Ω is the rotor speed and Ω_0 its steady state value. Further, $\mathrm{M}_{y\mathrm{T}}$ is the fore-aft tower base bending moment and $\mathrm{M}_{y\mathrm{T}0}$ its steady state value.

The following rules apply:

1. You can replace the feedback controller (ROSCO) and use your own combined feedback and feedforward controller, model predictive control etc. adjusting collective/individual pitch angle and generator torque.
2. Changes in the OpenFAST input files other than adjusting the BLADED INTERFACE section in the ServoDyn input file are not allowed.
3. Instead of the preview provided by the simple single point lidar (which moves with the nacelle), you can also use preview of the gust stored in a file or similar. The definition of the gust can be found in WriteGust2BladedWind.m.

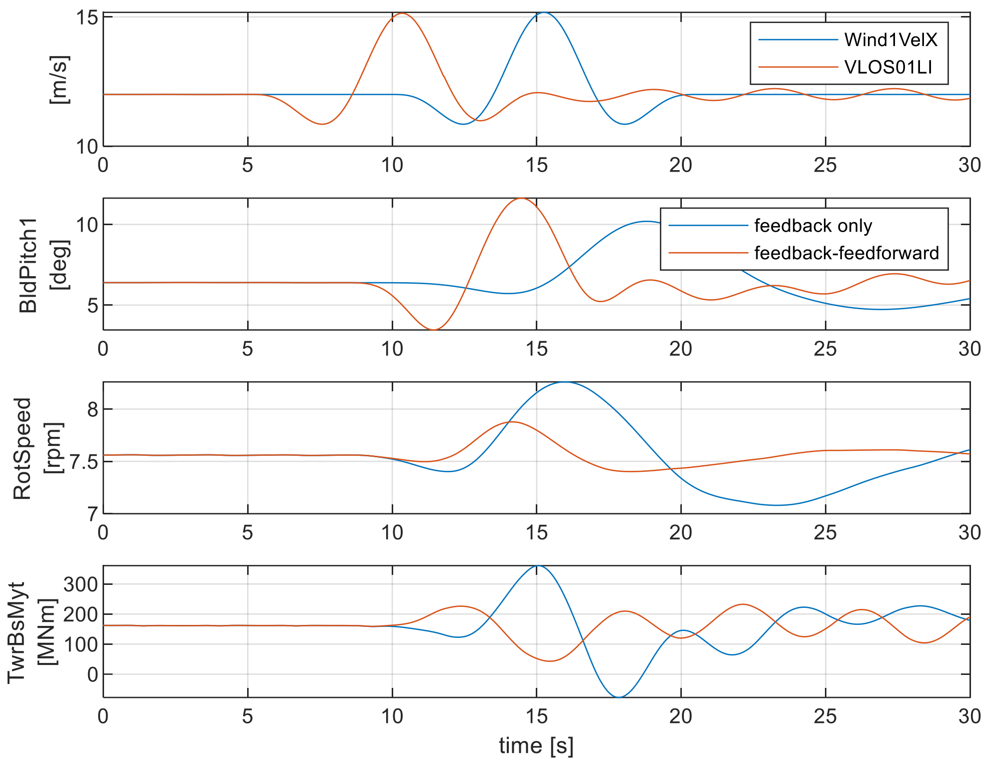

Figure 1: Starting point of "The 30 s sprint" challenge: Goal is to reduce the impact of a gust to rotor and tower motion. We would like to trigger new control concepts, since baseline collective pitch feedforward control provides room for improvements. Generated by the script RunExample.m in the Release/IEA15MW_01 folder.

The 18 m/s Hurdles

In the second challenge, we'll test how well your lidar-data-processing algorithm is able to detect the obstacles at high wind speeds, here 18 m/s. The challenge is mostly addressing lidar experts and the intention is to introduce you to lidar-data-processing for control and to trigger new lidar-data processing algorithms and lidar scan configurations. The objective is to reduce the measurement error of the predicted rotor-effective wind speed (REWS) with two seconds of prediction time, typically needed for feedforward control for this turbine.

The scripts RunExample_CircularCW/4BeamPulsed.m/.py in the Release/IEA15MW_03 folder provide examples using a baseline lidar-data processing [1] for a typical continuous wave lidar (circular scan) and a pulsed lidar (4 beams). To simplify the task, we ignore blade blockage, wind evolution, and other issues reducing the data availability.

The evaluation criteria for this discipline is the minimization of the mean absolute error (MEA) between the REWS from the wind field shifted two seconds into the future ($\mathrm{REWS_{WF}}$) and the processed REWS estimate from the lidar ($\mathrm{REWS_{LP}}$):

$$J = \mathrm{mean}(|\mathrm{REWS_{WF}} - \mathrm{REWS_{LP}}|)\ .$$

In the examples, the REWS is filtered by a low pass filter and then buffered. Due to historical reasons the filtering is realized in the FFP_v1.dll. The MEA is averaged over the last 600 seconds of six simulations with different turbulent wind fields generated by TurbSim.

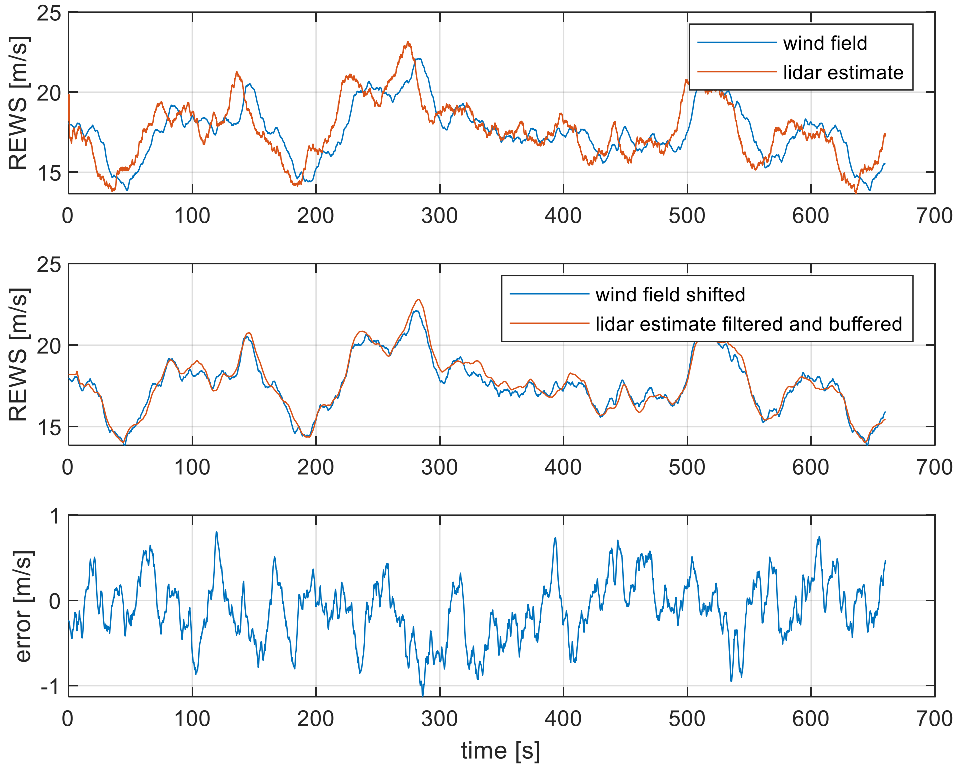

Figure 2: Starting point of "The 18 m/s hurdles" challenge: Goal is to reduce the measurement error (bottom). Here, the rotor-effective wind speed (REWS) from the wind field is shifted by two seconds into the future and compared to the processed lidar estimate (top and center). We would like to trigger new lidar data processing algorithms and scan configurations. Generated by the script RunExample_CirularCW.m in the Release/IEA15MW_03 folder.

The following rules apply:

1. The lidar configuration can be adjusted either by adjusting the lidar input files (LidarFile_CircularCW.dat and LidarFile_4BeamPulsed.dat) or by loading a new lidar input file (adjusting SWELidarFile in the OpenFAST input file). Only commercial available lidar systems can be used.
2. The lidar data processing can be used by adjusting the current DLL input files (FFP_v1*.DLL and LDP_v1*.IN) or by loading new DLLs by the Wrapper.dll (you can adjust the BLADED INTERFACE section in the ServoDyn input file, if you load another IN file for the Wrapper.DLL).
3. Changes to the input files other than the ones mentioned above or changes in the wind fields are not allowed.

The DLC 1.2 Marathon

Finally, the marathon. This discipline was included first during the 1896 Olympic games to honor Pheidippides, the Greek Messenger. Although we are not making you run the complete distance, we will ask you to test your controller performance by simulating the whole set of the Design Load Case 1.2 (DLC 1.2). The challenge is mostly addressing lidar-assisted control enthusiasts combining the knowledge from control and lidar-data processing. The intention is to benchmark full concepts and to identify the potential. The objective is maximizing the energy generated by a LAC controlled turbine either by increasing the Annual Energy Production (AEP) or by achieving Life Time Extension (LTE) while keeping important constraints.

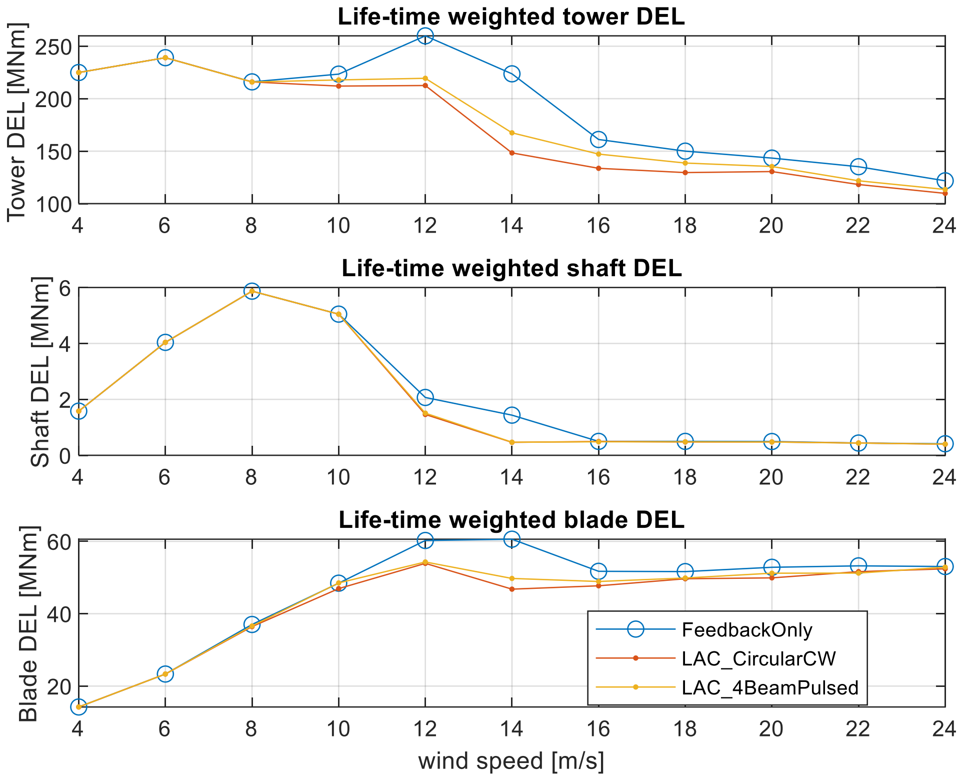

Figure 3: Starting point of "The DLC 1.2 Marathon" challenge: Goal is to increase the Annual Energy Production (AEP) or achieve Life Time Extension (LTE) by reducing tower and blade loads, keeping important constraints. We would like to benchmark full concepts and to identify the potential of LAC. Generated by the script CompareRuns.m in the Release/IEA15MW_05 folder.

For this more complex challenges, you need to perform the following steps from the Release/IEA15MW_05 folder:

1. Generate the 11 x 6 wind fields with GenerateTurbSimWindFields.m/.py
2. Run the feedback only simulations with RunExample_FeedbackOnly.m/.py.
3. Run the LAC simulations. Examples based on the previous two challenges are provided: RunExample_LAC_4BeamPulsed.m/.py and RunExample_LAC_CircularCW.m/.py.
4. Step 2 and 3 generate Statistic files (.mat or .csv), which are then evaluated with CompareRuns.m/.py.

The evaluation criteria for this discipline is the maximization of the ratio between the energy produced by the LAC controlled turbine (annual energy produced $\mathrm{AEP_{LAC}}$ times the LTE, which is the minimum of the LTE of tower and blades) and the conventional controlled turbine ($\mathrm{AEP_{FB}}$):

$$J = \frac{\mathrm{LTE}\times \mathrm{AEP_{LAC}}}{\mathrm{AEP_{FB}}} \text{ with } \mathrm{LTE} = \min\left(\mathrm{LTE_{Tower}}, \mathrm{LTE_{Blades}}\right).$$

Details on how to calculate the LTE from the Damage Equivalent Loads (DEL), see Figure 3, can be found in [5], and code is provided in the repository to calculate the LTE. We are well aware that this is very simplified, since other components etc. are neglected, but we would like to keep it as simple as possible at this stage.

Further, the following constraints need to be kept (1-4 in relation to feedback only).

1. The energy increase cannot be reduced.
2. Tower and shaft loads cannot be increased.

3. Maximum over-speed cannot be increased.
4. The life-time weighted pitch travel cannot be increased.
5. The maximum generator torque limit needs to be kept.

The following rules apply:

1. You can replace the feedback controller (ROSCO) and use your own combined feedback and feedforward controller, model predictive control etc. adjusting collective/individual pitch angle and generator torque.
2. You can replace the lidar configuration and lidar data processing (see previous challenge).
3. Changes to the input files other than the ones mentioned above or changes in the wind fields are not allowed.

Our timeline

The Summer Games 2024 have started on April 3 and will last until June 30.

What should be provided:

1. Necessary: Simulation results: please upload the zipped outb files to the Teams folder Uploads (WG 2 – LAC>SummerGames2024>Uploads) in the corresponding IEA15MW0x folder using the following filename for your zipped file: <Name>_<YYYYMMDD>, see example WETI_20240330.zip. For the discipline “The 18 m/s Hurdles” you also need to upload files with your REWS estimate from your lidar data processing algorithm (similar to the ROSCO dbg files). For the other two disciplines, please only upload outb files with the same structure and filenames as in WETI_20240330.zip files.
2. If possible: In addition to the files above, DLLs and other files necessary to run simulations (via a Git fork). Please check with someone independently to ensure that the scripts run and state clearly, which script produces the submitted files.
3. If wished: Source Code of your DLLs can be committed in a Git Fork and used for highlighting the improvements.

We will then evaluate the results together in our working group meetings and announce the rankings in the three disciplines and two categories.

Any changes to the rules or submission dates can be discussed in the discussion forum: https://github.com/IEAWindTask52/LidarAssistedControl/discussions.

How to get started

To get started with our Lidar-Assisted Control open-source tools, you need to clone the GitHub repository in your local computer using either Git or GitHub Desktop. Detailed information about the DLLs and how you can compile them is given in the file README.MD. In the repository we have given various example scripts for LidarAssistedControl in both Matlab and Python.

To get started with the Matlab scripts you need to navigate to the ‘Release’ folder and then you can start working based on the example scripts (see disciplines above). The scripts use functions which are provided in a folder WetiMatlabFunctions and added to the MATLAB path.

To get started with Python scripts unlike Matlab users need to install various modules like numpy, pandas, or scipy. To make it simpler we have provided a 'setup.py' script that installs all the necessary modules in your environment.

Support

If you have questions regarding the Summer Games in general or regarding the code which might be interesting for others as well, please use our Forum on GitHub: https://github.com/IEAWindTask52/LidarAssistedControl/discussions.

If you require further support, please don't hesitate to contact via email our support team:

- Simon Weich - simon.weich@stud.hs-flensburg.de
- Aravind Venkatachalapathy - aravind.venkatachalapathy@stud.hs-flensburg.de

References

[1] Guo, F., Schlipf, D., and Cheng, P. W.: Evaluation of lidar-assisted wind turbine control under various turbulence characteristics, Wind Energ. Sci., 8, 149–171, DOI: 10.5194/wes-8-149-2023, 2023.

[2] Schlipf, D., Guo, F., Raach, S., Lemmer, F.: A Tutorial on Lidar-Assisted Control for Floating Offshore Wind Turbines, American Control Conference, DOI: 10.23919/ACC55779.2023.10156419, 2023.

[3] Gaertner, E., Rinker, J., Sethuraman, L., Zahle, F., Anderson, B., Barter, Garrett E., Abbas, N. J., Meng, F., Bortolotti, P., Skrzypinski, W., Scott, G. N., Feil, R., Bredmose, H., Dykes, K., Shields, M., Allen, C., Viselli, A.: IEA Wind TCP Task 37: Definition of the IEA 15-Megawatt Offshore Reference Wind Turbine. DOI: 10.2172/1603478, 2020.

[4] Abbas, N. J., Zalkind, D. S., Pao, L., and Wright, A.: A reference open-source controller for fixed and floating offshore wind turbines, Wind Energ. Sci., 7, 53–73, DOI: 10.5194/wes-7-53-2022, 2022.

[5] Schlipf, D. , Fürst, H., Raach, S., Haizmann, F., Systems Engineering for Lidar-Assisted Control: A Sequential Approach, WindEurope conference, Hamburg, Germany, September 2018, Journal of Physics: Conference Series, DOI: 10.1088/1742-6596/1102/1/012014.