Design, Analysis, and Learning Control of a Robotic Wind Turbine

J. Zico Kolter, Zachary Jackowski, Russ Tedrake*

***Abstract*— Wind power represents one of the most promising sources of renewable energy, and improvements to wind turbine design and control can have a significant impact on energy sustainability. Although wind energy has not been a canonical research area in robotics, we argue here that many robotics techniques (physical system design, modeling and control, and policy learning techniques) can in fact have a large impact on wind energy research. Pursuing this goal, in this paper we develop a small, fully functional robotic wind turbine, capable of variable speed operation and individual pitch actuation. We analyze the power output of the turbine under different operating conditions, comparing to predictions from aerodynamic simulators. Using inverse dynamics control we demonstrate high-frequency blade pitch control synchronized to rotor angle. Finally, using a novel Reinforcement Learning policy search algorithm, we show that the system can quickly optimize its energy output in an online and fully model-free manner.**

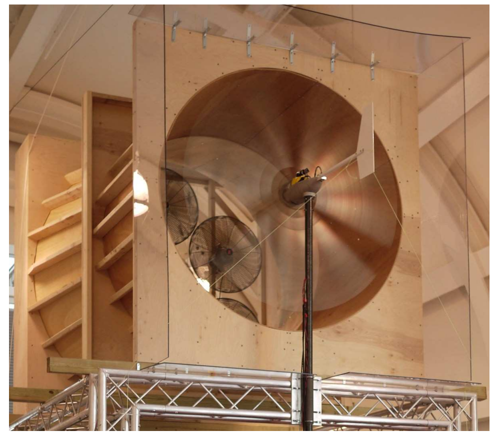

Fig. 1. Picture of our micro wind turbine in operation.

I. INTRODUCTION

Energy issues pose one of the greatest challenges facing society. More than 86% of the world's energy currently comes from (unsustainable) fossil fuels, and worldwide energy demand continues to grow rapidly [1]. Wind power represents one of the most promising sources of renewable energy: currently wind is more economically feasible than solar or biomass for electricity generation, with some projections predicting wind, given good environmental conditions and proper government subsidies, to be of similar cost to fossil fuels for electricity generation [2]. Despite this promise, significant improvements in the deployment and control of wind turbines are needed if wind is to contribute a significant portion of electricity worldwide [3].

Although wind energy has not historically received much attention in robotics, we argue here that the robotics community as a whole has a great deal to offer to wind research. At this point the purely aerodynamic properties of wind turbines are well understood, and the challenge is to extract maximum power (or to minimize adverse structural loads) under unsteady and uncertain operating conditions, a common theme in much recent work in robotics. Thus, the control and modeling techniques developed in robotics have the potential to greatly influence how wind turbines are deployed. Furthermore, despite the rise in wind power installations, there are remarkably few physical systems dedicated to research work, and most current research in wind power is conducted entirely in simulation. This is another facet where robotics can have a large impact: by designing small and low-cost systems for research work, robotics has the opportunity to greatly accelerate research in the wind community as a whole.

In this paper, we present first steps in bridging wind and robotics research. We have developed a micro wind turbine (which we call the Turbot, for turbine robot, pictured in Figure 1) with individual blade pitch control, variable rotational speed and power output, all controlled by an external computer; we describe the design of this system and its software in Section III. In Section IV-A we then analyze the power output of the system under different operation conditions, and compare this to the power predicted by turbine simulation software developed at the National Renewable Energy Laboratory; the results show that overall the turbine behaves qualitatively as expected, but with significant differences that underscore the value of testing on a real system versus simulation alone. We demonstrate the individual pitch control mechanism of the turbine in Section IV-B, highlighting the need for inverse dynamics control to achieve accurate tracking. Finally, in Section IV-C we apply an extension of recent work in policy search Reinforcement Learning to autonomously optimize power output of the turbine in a model-free, online manner.

II. BACKGROUND AND RELATED WORK

Wind energy has received a great deal of attention in recent years, and there is a vast amount of work on the design, modeling, and control of wind turbines. A full review of relevant literature is beyond the scope of this paper, and we focus instead only on those elements of turbine design and control that are most relevant to the presentation here. However, there are several introductory texts on wind energy in general, and we refer the reader to these for more

*All authors are with the Computer Science and Artificial Intelligence Laboratory, Massachusetts Institute of Technology, Cambridage MA, 02139 {kolter,zackbass,russt}@csail.mit.edu.

information [4], [5]. A number of papers survey more recent control work and challenges [6], [7].

The two trends in turbine controls work that relate most directly to this paper is work on individual pitch control (IPC) and maximum power optimization. While many control strategies use collective pitch adjustments to change the behavior of the turbine, a number of recent approaches have focused on individually pitching blades, typically to decrease adverse structural loads [8], [9], [10]. This control is often performed by transforming the system to a angle-independent coordinate frame via the so-called "d-q" or Coleman transformations [8], [11], [9], linearizing a steady-state model in this coordinate frame, and then converting the resulting inputs back to actual pitch commands. Alternative work considers periodic control directly [12], [13], but we note that the final output of both these systems is largely periodic blade commands, which is the focus on the control strategies for IPC in this paper.

The second piece of related work is research on power optimization using collective pitch control, which is precisely the problem we focus on using policy search techniques. Most initial work in this area was largely model based, and focused on methods for accurately tracking the optimal control setpoints under changing conditions [14], [15], [16]. Alternatively, newer work has focused on model free-methods for optimizing power coefficients, using extremum seeking control [17] or finite difference methods [18]. These approaches are conceptually similar to the proposed approach here, but are strictly first-order methods, do not efficiently reuse past trajectories, and were tested only in simulation.

Since this paper is about a physical platform, we also briefly discuss turbine controls work on real hardware. The vast majority of research in advanced wind control techniques (including all papers cited above) is performed entirely in simulation: the ubiquity of standard simulation packages such as FAST [19] and WT_Perf [20], as well as the cost involved with work on a a full-scale turbine, have made full system real-world experiments a rarity in wind research. Notable exceptions to this include work on the Controls Advanced Research Turbine (CART) [21], including work on load reduction [22] and individual pitch control [23]. However, work on the CART has focused largely on model-based approaches, largely due to the fact that significant effort has been invested in ensuring accurate models for this particular turbine. We also know of some work on smaller scale turbines [24], but know of no such systems with active pitch control; indeed, we would argue that the hardware needed for individual pitch control on a micro turbine, for example, has only become available on a large scale in recent years.

A. *Motivating Challenges*

Although we mentioned in the introduction that wind power involves challenging control tasks with many open question, we want to close by briefly elaborating upon some of the control and design challenges posed by modern wind turbine systems. We make no claims to having solved these problems in the current paper, but they have motivated much of our current work in wind energy, and they highlight the richness and complexity of the control challenges involved. Our hope is that these challenges, along with the initial systems presented in this paper, will motivate the robotics community as a whole to take on some of these problems.

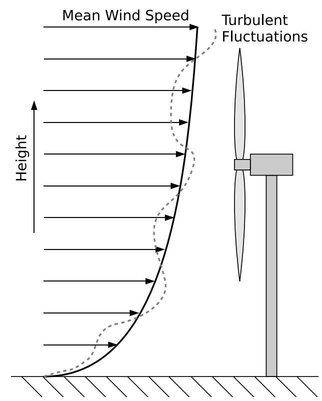

Fig. 2. Illustration of wind shear and turbulent effects on a wind turbine. Based upon [25].

Our first example highlights the inherent complexity of wind power even in a seemingly simple situation, a single turbine sitting on even terrain. Modern wind turbines are massive constructions, with rotor diameters of over 100 meters [26], and this introduces a variety of unsteady aerodynamic conditions. Wind speed increases with height above ground level, an effect known as wind shear, pictured in Figure 2. For large turbines, the result is that the wind at the top of the turbine is moving much faster than the wind at the bottom, often times necessitating a changing control strategy within one blade revolution; indeed such asymmetric forces were one of the original motivations for individual pitch control. Added to this general trend in mean wind speed are turbulent fluctuations and gusts [25], some operating within the time scale of a single revolution and some persisting for multiple revolutions. Given these variations, the optimal control strategy for a single turbine operating under seemingly simple conditions can be quite complex.

Second, we note that single turbine installations are becoming rather uncommon: most commercial wind energy sites now house wind farms, collections of many turbines operating in close proximity. The adverse aerodynamic conditions are typically only exacerbated in such situations: the blades of upwind turbines shed vortices that mix with the incoming flow and the turbine towers themselves create turbulent wakes that spread to downwind turbines. Figure 3 shows an illustration of wakes effects on a wind farm, as well as a photograph showing such interactions in the Horns Rev offshore wind farm. The end result of these conditions is that most wind farms space turbines very far apart to mitigate the risks as much as possible, but still feel these effects [28]. Any improvements in control or design that can enable turbines to be moved closer together can result in the

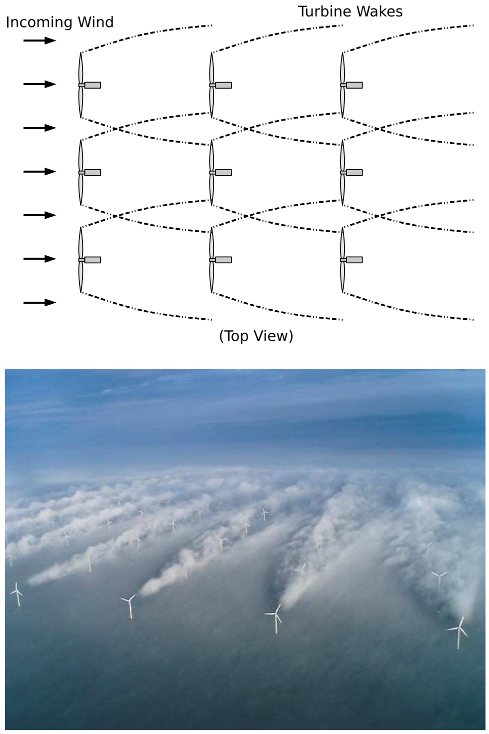

Fig. 3. (Top) Illustration of wake effects on a turbine farm, based upon [25]. (Bottom) Photograph from Horns Rev 1 offshore wind farm [27], illustrating wake effects in a physical system.

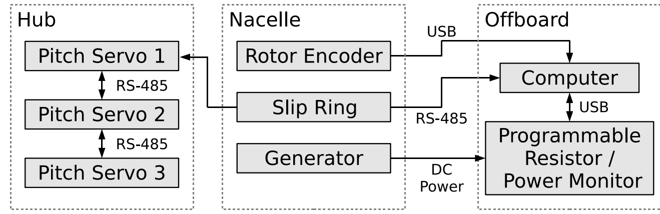

Fig. 4. Block diagram of the hardware components in the Turbot.

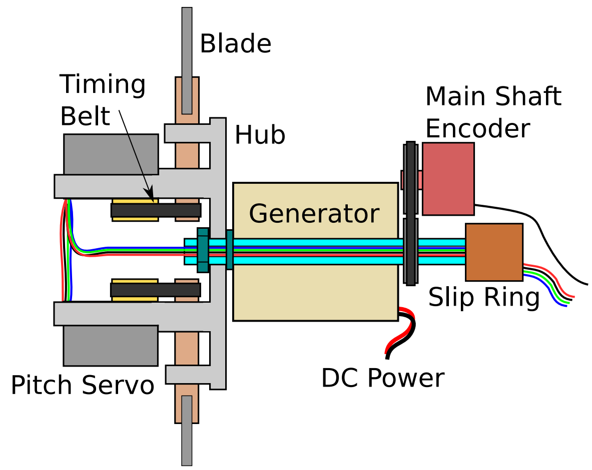

Fig. 5. Mechanical diagram of the turbine hub and nacelle.

capture of significantly more wind energy.

With these problems as general motivation, we now present our robotic turbine platform, along with analysis, control, and policy search methods that employ techniques based upon robotics and learning methods to control and optimize this system.

III. HARDWARE AND SOFTWARE DESIGN

The Turbot is a fully functional wind turbine, with a blade radius of 1.52 meters and a maximum power output of 300W DC. The system is based upon the Extractor turbine designed by Alternate Power Technologies, Inc.[1], but with significant modifications. We replaced the mechanical blade pitch mechanism of the Extractor turbine with servo motors attached to each blade root; this allows us to individually control the turbine blades at very high frequencies. We transmit data and power signals to the servos using a slip ring in the turbine nacelle and track the rotor angle using an encoder off the main rotor shaft. Offboard, the power generated by the system is fed into a board that both monitors the power output by the system, and can programmably vary its resistance (this in turn produces more or less torque on the generator). Finally, a simple wind tunnel built using commodity fans powers the turbine itself. All the elements are controlled by an offboard computer via the Lightweight Communication and Marshaling (LCM) software [29], a message-passing framework developed for the DARPA urban challenge. A block diagram of these components is shown in Figure 4.

One element we want to highlight is the total cost of the system, which is approximately $3,250 for the turbine and associated hardware ($1,000 for the Extractor turbine, $1,550 for the Dynamixel servos, $350 for encoder and slip ring, $250 for the power monitoring hardware and $100 for various cables and wiring) plus $2,600 for the wind tunnel ($2,100 for fans, and $500 for lumber). We are not aware of any low-cost micro turbines that have the capabilities of our system (individual pitch control, variable speed operation); thus, we feel the platform itself has the potential to significantly impact small and academic-scale research in wind power. We are happy to share detailed design specifications with anyone wishing to build a similar system.

A. Hub and Nacelle Design

The turbine hub and nacelle house the pitch actuators, rotor encoder, and generator, as shown in Figure 5. To control blade pitch, we use three Dynamixel EX-106+ servo motors[2] attached to the blade roots. The servos are daisy chained together and controlled digitally via an RS-485 link, which allows for setpoint control at 250 Hz with less than 1ms latency (in addition to the onboard PD controller in the servos). A picture of the hub and servo is shown in Figure 6.

[1] http://www.vpturbines.com

[2] http://www.robotis.com/xe/dynamixel_en

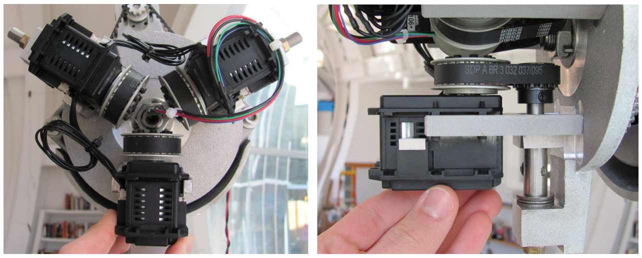

Fig. 6. Front view of the turbine hub and a closeup of the servo motor used for pitch control.

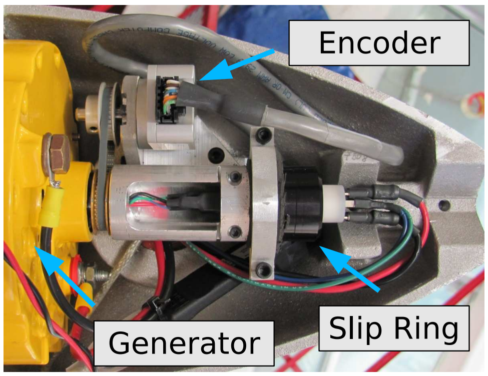

Fig. 7. Internals of the turbine nacelle, showing the generator, encoder and slip ring.

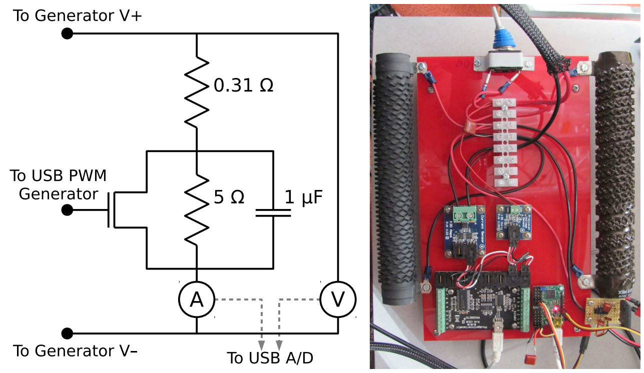

Fig. 8. Circuit diagram (left) and photograph (right) of the programmable resistor and power monitoring board.

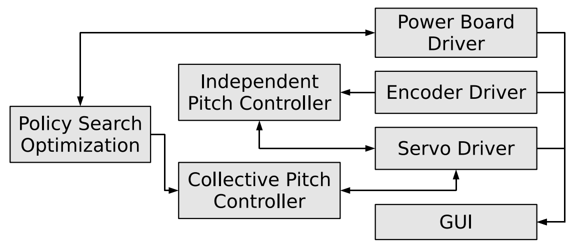

Fig. 9. Block diagram of the software architecture for controlling the Turbot.

Since the hub itself rotates, to provide data and power to the servos we use a Mercotac 430 slip ring[3] in the nacelle, and run the wires through the rotor shaft. The nacelle also houses a US Digital HB5M encoder[4] off the main shift used to track the orientation of the rotor (necessary both to estimate speed and to synchronize blade orientation with rotor angle) as well as the generator itself, a Delco 12SI alternator modified for use with the Extractor turbine. The components of the nacelle are shown in Figure 7.

B. Power Control and Monitoring

To monitor and regulate the power output by the turbine we attach a resistor to the DC power output lines, and monitor current and voltage using a Phidgets USB A/D converter and associated voltage and current measuring devices.[5] Inn order to regulate the power output it is important to be able to vary the effective resistance: lower resistance will impose a larger torque on the generator, and some intermediate (but unknown) resistance is needed to maximize power output. Thus, we place two high-power resistors (0.31 and 5 Ohm respectively) in series, and short the later with a MOSFET controlled by a PWM signal at 25kHz. By varying the duty cycle of the PWM from 0 to 1, we can smoothly interpolate between a resistance of 5.31 and 0.31 Ohms. The circuit diagram for this device, as well as a photograph of the board, is shown in Figure 8.

[3]`http://www.mercotac.com`
[4]`http://www.usdigital.com`
[5]`http://www.phidgets.com`

C. Software Architecture

The turbine is controlled via an offboard computer, running a software system built upon the LCM message passing framework. LCM offers an attractive architecture for building such a system, as the different drivers and controllers can be developed in a modular fashion, and the system has built-in logging and playback capabilities. The basic software architecture is shown in Figure 9. In addition to the three driver modules that manage the servos, the encoder, and the power board, we use two basic controllers: 1) a collective pitch that controls all the blades equally, using a simple integral control loop (although the servos internally have a PD controller, the friction in the drive train and steady aerodynamic forces often cause these to have a steady state error, which we correct via an integral controller), and 2) an independent pitch controller that synchronizes blade pitch of the different blades with rotor angle using inverse dynamics; this element will be discussed further in Section IV-B. Also illustrated in Figure 9 is a policy search procedure that we will describe in Section IV-C, which uses collective pitch control and feedback from the power board to optimize energy output, and a MATLAB graphical user interface that allows for easy visualization and control of the different modules.

D. Wind Tunnel

To power the turbine, we constructed a small circular wind tunnel using wood and commodity commercial fans. The tunnel is eight feet long (a four foot contraction section and

Fig. 10. Rear view of the wind tunnel used for powering the Turbot.

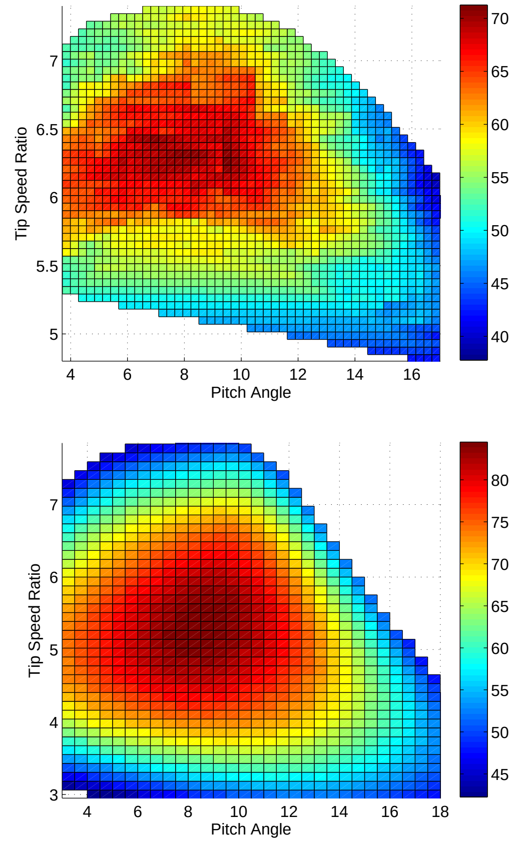

Fig. 11. Real (top) and simulated (bottom) power curves for the Turbot under different operating conditions. Power levels indicated by color are in Watts.

a four foot straight section), with an output diameter of 1.72 meters (20 cm larger than the diameter of the turbine), and generates an average wind speed of 6.5 m/sec over this area (approximately 13.5 miles per hour). This is not fast enough to reach the rated power output of 300W, but is still able to generate a significant amount of power at high rotational speeds. A photograph of the fans powering the tunnel is shown in Figure 10.

IV. ANALYSIS AND RESULTS

A. Modeling and Power Output Analysis

Our first set of experiments characterize the power output of the Turbot relative to a simulated model of the system, under a variety of different operating conditions. Specifically, we independently varied the (collective) blade pitch angle of the turbine from 4 to 17 degrees in 24 equal increments, and the resistance from 1.8 to 5 Ohms in 14 equal increments, for a total of 336 different operating conditions. For each setting, we let the turbine run for five seconds (to settle to the steady state power output), and recorded the resulting power.

In addition, we developed a simulated model of the Turbot system using the WT_Perf aerodynamics simulator [20], an industry standard tool developed by the National Wind Technology Center for predicting the the performance of wind turbines using a technique known as blade element momentum (BEM) theory [4]. A full description of the WT_Perf simulator or BEM theory is beyond the scope of this paper, but briefly, the simulator takes as input the physical properties of the system such as the number of blades and their shape, the lift and drag curves of the blade airfoil, and aerodynamic constants such as the density and viscosity of the air. It then computes the aerodynamic properties of the blades, along with momentum-based properties of the turbine power output as a whole, to compute steady-state power output for different operating conditions of the turbine.

The resulting power curves, for both the real and simulated systems, are shown in Figure 11. The figures show power output in Watts, indicated by color, as a function of pitch angle and *tip speed ratio*, the ratio of the blade's tip speed to the incoming wind speed; although this may seem a somewhat unusual way of describing the operating conditions of the turbine (since the actual control inputs are blade pitch and generator resistance, both of which affect the rotation speed and hence tip speed ratio), describing the power output as a function of tip speed ratio is the standard in the wind literature for a number of reasons: the coefficient of power (the ratio of captured to available power) can be roughly approximated as a function of blade pitch and tip speed ratio independent of wind velocity, and a fixed generator torque and pitch angle can potentially have two different equilibrium points in terms of the rotor speed.

Figure 11 also emphasizes both the the benefits and drawbacks of simulation models. Qualitatively, the two power curves look quite similar: they both have a characteristic "triangle" shape, and the optimal operating conditions are at similar points. But the figures also differ in crucial respects: the optimal tip speed ratio on the real system is significantly higher than for the simulated system, and the range of near-optimal points is significantly smaller. Indeed, if one were to simply choose the operating conditions of the turbine based upon the simulated model (as is often done in practice), then we would be operating at a substantially suboptimal point. This highlights the usefulness of experimentation on real systems and the development of learning methods that can optimize performance without relying on an a priori model of the system, both areas where the robotics community has a great deal of experience.

B. Independent Pitch Control

Our second set of experiments demonstrate the feasibility of independent pitch control (IPC) on the Turbot. Independent pitch control was a crucial design requirement for our system since, as discussed in Section II, many modern control approaches exploit individual pitch control. Yet IPC on a micro turbine is a challenging task: in order to maintain reasonable tip speed ratios, micro turbines must rotate much faster than large turbines (the maximum power output from the previous section occurs at a tip speed ratio of 6.25, corresponding to a rotor speed of 510 RPM on the Turbot). IPC requires at a minimum that we be able to vary the blade pitch on the order of one rotational period, typically synchronized with the absolute rotor angle in order to account for spatial effects such as wind shear. Thus, we need both high frequency control and accurate compensation for the servo motor dynamics in order to accurately track the desired pitch angles.

To achieve accurate independent pitch control, we employ an inverse dynamics control system. The dynamics of the motors are well-modeled by a second order system

$$\ddot{q} = k_1(q - u) + k_2\dot{q} + k_3\mathrm{sign}(\dot{q}) + k_4 \tag{1}$$

where q denotes the motor angle, u denotes the control input (corresponding to the desired angle setpoint), $k_1, \ldots, k_4$ are parameters of the system, and where the system is non-linear due to the sign term, which captures the effects of Coulomb friction. We fit this model to the data using system identification procedures: we generate data by commanding a sequence of "chirp" commands (sinusoidal inputs and varying frequencies and magnitudes), then minimize the *simulation error* of the model (the deviation between the observed and predicted sequences, simulated over the entire sequence of inputs) using non-linear optimization.

Suppose now that we want to command the blade pitch q to be some known function of the rotor angle θ

$$q^d = f(\theta). \tag{2}$$

We analytically differentiate this function to achieve desired velocity and accelerations

$$\dot{q}^d = f'(\theta)\dot{\theta}, \quad \ddot{q}^d = f''(\theta)\dot{\theta} + f'(\theta)\ddot{\theta} \tag{3}$$

and where we will typically assume for simplicity that the rotor speed is constant $\ddot{\theta} = 0$. Now, instead of simply commanding the desired angle $u = q^d$, we invert the dynamics model and command the input

$$u = q^d + (\ddot{q}^d - k_2\dot{q}^d - k_3\mathrm{sign}(\dot{q}^d) - k_4)/k_1 \tag{4}$$

which can be interpreted as adding a feedforward compensator to the desired position, based upon the dynamical model.

Figure 12 shows the pitch tracking performance using the inverse dynamics versus simple PD control. We commanded a desired angle of $q^d(\theta) = 12 - 2.9\sin(\theta)$, with the rotor spinning at 525 RPM. The top plot of Figure 12 shows tracking performance of the inverse dynamics controller: in this case we are able to track the reference trajectory quite accurately even at high speeds, with an RMSE of 0.26 degrees. In contrast, the bottom plot shows the performance of a simple PD controller attempting to track the trajectory; here the settling time of the motor dynamics cause the actual trajectory to lag significantly behind the desired trajectory, a well-known effect in PD control. While it would also be possible to adaptively tune the phase and amplitude of the desired trajectory to make the PD controller match the desired behavior, the inverse dynamics approach essentially does this automatically through its model of the dynamics.

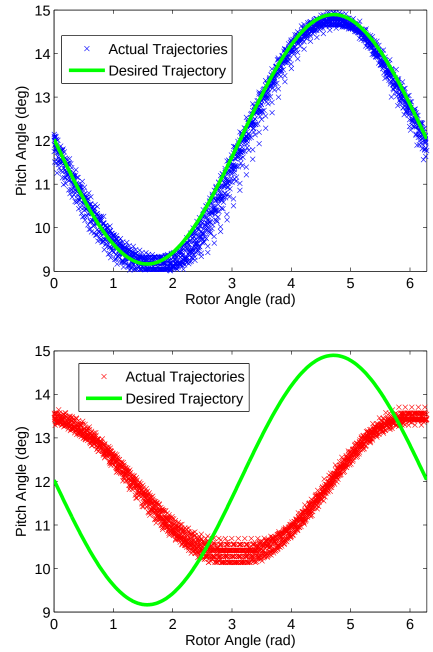

Fig. 12. Individual Pitch Control tracking performance using Inverse Dynamics (top), and PD (bottom) control.

C. Policy Search for Online Parameter Optimization

Finally, our last experiments on the turbine return to collective pitch control, but demonstrate the promise of learning and adaptive control approaches for optimizing performance. The general approach is motivated by the fact that, as shown in Section IV-A, choosing the optimal operating conditions for the turbine based purely on a model can lead to highly suboptimal plans. Assuming that we typically would not want to sweep out the entire parameter space for most policies (especially as the dimension of the policies grows), a natural alternative is to use *policy search* Reinforcement Learning methods [30], [31] to optimize policy parameters online. Specifically, we will apply policy search to optimize the operating conditions of the turbine (the pitch angle and tip speed ratio or resistance) to maximize power output in an entirely online fashion, without a model of the system.

Our approach is based upon recent work in policy gradient techniques, notably the REINFORCE algorithm [30] and work on importance sampling in this domain [32], but includes significant modifications that make it perform must better in our domain. Let $R(w)$ denote the reward for policy parameters $w \in \mathbb{R}^n$ (in this case the pitch angle and load resistance of the turbine). Our method is based upon computing the second order Taylor approximation to this function

$$R(w) \approx R(w_0) + g^T(w - w_0) + \frac{1}{2}(w - w_0)^T H(w - w_0) \quad (5)$$

where $g \in \mathbb{R}^n$ and $H \in \mathbb{R}^{n \times n}$ denote the gradient and Hessian of $R(w_0)$ respectively. Of course, since the reward is unknown without a model of the dynamics, we cannot compute these terms analytically. But, given m samples of the reward for a number of different parameters $w^{(i)}$ we can compute them in a least squares sense

$$\begin{bmatrix} \phi(w^{(1)} - w_0)^T \\ \vdots \\ \phi(w^{(m)} - w_0)^T \end{bmatrix} \begin{bmatrix} \mathrm{vec}(H) \\ g \\ R(w_0) \end{bmatrix} \approx \begin{bmatrix} R(w^{(1)}) \\ \vdots \\ R(w^{(m)}) \end{bmatrix} \quad (6)$$

where

$$\phi(w^{(i)} - w_0) \equiv \begin{bmatrix} \mathrm{vec}((w^{(i)} - w_0)(w^{(i)} - w_0)^T) \\ w^{(i)} - w_0 \\ 1 \end{bmatrix} \quad (7)$$

is the vector of all first and second order terms appearing in the expansion and vec denotes the vectorization of a matrix. This approximation is analogous to finite differencing [31], except that we are computing a second order approximation to the function. Given this approximation, we employ a trust region optimization strategy, which selects the next iterate of w as

$$w_{t+1} \leftarrow w_t + \arg \max_{\Delta w^T Q \Delta w \leq 1} \frac{1}{2} \Delta w^T H \Delta w + g^T \Delta w \quad (8)$$

where $Q \in \mathbb{R}^{n \times n}$ defines the trust region. Although this problem is non-convex when H is not negative definite (for instance if we are not at a local optimum, or if we do not have sufficient samples to obtain an accurate least-squares estimate), it can still be solved exactly using a semidefinite relaxation [33, Appendix B]. Most numerical trust-region methods avoid solving this subproblem exactly [34], as it can itself by quite time consuming, but we argue that policy search is a natural fit for exact trust region methods: since the dimensionality of the problem is typically relatively small, and since the computational time is typically much less than the execution time on the real system, exact trust region optimization is well-suited to this task.

To select the parameters where we sample the reward function, we follow standard practice of most RL policy search methods, and sample parameters according to a stochastic policy $w_t + \epsilon_t$, where

$$\epsilon_t \sim \mathcal{N}(0, \Sigma_t) \quad (9)$$

and the covariance Σ_t is an input parameter to the overall policy search procedure, typically chosen to be diagonal or even isotropic. In order to reuse samples from past iterations

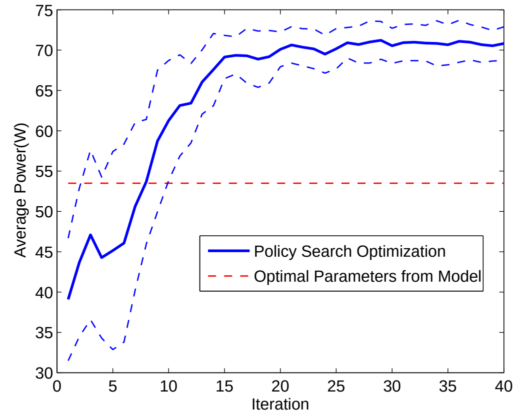

Fig. 13. Average reward versus iteration number, with 95% confidence intervals for our trust region policy search method optimizing power output.

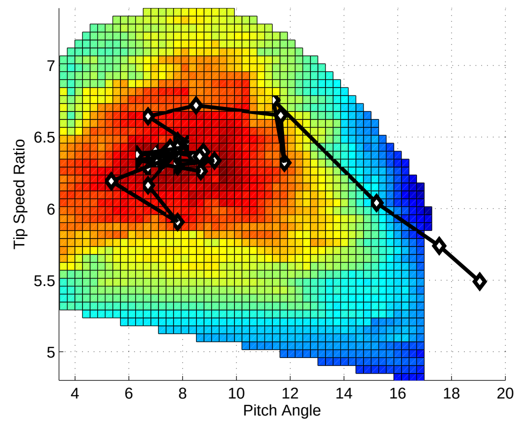

Fig. 14. Evolution of policy parameters for a typical run of policy search.

(particularly important given that we are estimating a full Hessian), we use importance weighting as described in [32], and solve a *weighted* least squares problem to estimate the Hessian; in our final implementation we sample only one set of policy parameters at each iteration, and are still able to obtain accurate estimates of the cost surfaces. Finally, we propose to use $Q = \Sigma_t^{-1}$ as the trust region, since roughly speaking this requires that parameter updates lie within the same region as our typical parameter exploration.

To evaluate our policy search approach, we randomly initialized policies with arbitrary pitch angles and resistance values within some range. We then ran our trust region policy search approach for 40 iterations, using the power output averaged over two seconds (after three seconds of settling time) as the reward signal. Figure 13 shows the evolution of the reward versus iteration number, averaged over 10 trials (with different starting parameters), along with 95% confidence intervals. Also shown is the average reward obtained by trusting the WT_Perf simulation model and using its prescribed optimal operating point. Despite receiving very little feedback from the system, just the average power for a small number of situations without any model of the system, our algorithm is able to quickly obtain near-optimal parameter settings, typically within about 15 iterations (75 seconds of real-time operation). Figure 14 shows the evolution of the policy parameters for a typical run of the policy search; as expected the method quickly gets to (and remains in) a near-optimal region of the parameter space.

V. CONCLUSIONS AND FUTURE WORK

This paper has presented a robotic wind turbine system, with a focus on controlling and optimizing the system using techniques familiar to the robotics community: physical system design, inverse dynamics control, and model-free Reinforcement Learning policy search. Although the methods can attain impressive performance on the tasks we consider, we fully admit that the challenges posed by wind energy go well beyond what we directly consider in this paper. Next steps for the research involve integrating the policy search and individual pitch control mechanisms to cope with static obstructions in the incoming airflow. Building upon this, the eventual goal is to develop a system that can autonomously adjust to dynamic disturbances, such as those caused by an upwind turbine, using a combination of both model-based and model-free optimization techniques.

ACKNOWLEDGMENTS

We thank Morgan Quigley for helpful discussions, Paul Stearns of Alternate Power Technologies, Inc for the Extractor turbine and his assistance in modifying hub parts for our turbine, and Ron Wiken for his help designing and building the wind tunnel. J. Zico Kolter is supported by an NSF Computing Innovations Fellowship.

REFERENCES

[1] Various, "Annual energy review 2009," U.S. Energy Information Administration, Tech. Rep., 2009.
[2] ——, "Annual energy outlook 2011," U.S. Energy Information Administration, Tech. Rep., 2010.
[3] U. D. of Energy, "20% wind energy by 2030: Increasing wind energy's contribution to u.s. electricty supply," U.S. Department of Energy, Tech. Rep., 2008.
[4] J. Manwell, J. McGowan, and A. Roberts, *Wind Energy Explained: Theory, Design, and Application*. Wiley, 2009.
[5] T. Burton, D. Sharpe, N. Jenkins, and E. Bossanyi, *Wind Energy Handbook*. Wiley, 2001.
[6] T.K.Barlas and G. van Kuik, "Review of state of the art in smart rotor control research for wind turbines," *Progress in Aerospace Sciences*, vol. 46, no. 1, pp. 1–27, 2010.
[7] L. Y. Pao and K. E. Johnson, "Control of wind turbines: Approaches, challenges, and recent developments," *IEEE Control Systems Magazine*, vol. 31, pp. 44–62, 2011.
[8] E. Bossanyi, "Individual blade pitch control for load reduction," *Wind Energy*, vol. 6, pp. 119–128, 2003.
[9] T. Larsen, H. Madson, and K. Thomson, "Active load reduction using individual pitch, based on local blade flow measurements," *Wind Energy*, vol. 8, pp. 67–80, 2005.
[10] K. Selvam, "Individual pitch control for large scale wind turbines: Multivariable control approach," Energy research Centre of the Netherlands, Tech. Rep., 2007.
[11] E. Bossanyi, "Further load reduction with individual pitch control," *Wind Energy*, vol. 8, pp. 281–485, 2005.
[12] A. Wright, "Modern control design for flexible wind turbines," National Renewable Energy Laboratory, Tech. Rep., 2004.
[13] K. Stol, "Disturbacne tracking and blade load control of wind turbines in variable speed operations," National Renewable Energy Laboratory, Tech. Rep., 2003.
[14] K. Pierce, "Control methods for improved energy capture below rated power," in *Proceedings of the ASME/JSME Joint Fluids Engineering Conference*, 1999.
[15] J. Youqin, Y. Zhongging, and C. Binggang, "A new maximum power point tracking control scheme for wind generation," in *Proceedings of the IEEE International Conference on Power Systems Technology*, 2002.
[16] E. Koutroulis and K. Kalaitzakis, "Deisng of a maximum power tracking system for wind-energy conversion applications," *IEEE Transactions on Industrial Electronics*, vol. 53, no. 2, pp. 386–394, 2006.
[17] T. Hawkins, W. White, G. Hu, and F. D. Sahneh, "Wind turbine power capture control with robust estimation," in *Proceedings of the Dynamic Systems and Control Conference*, 2010.
[18] J. Creaby, Y. Li, and J. Seem, "Maximizing wind turbine energy capture ssing multi-variable extremum seeking control," *Wind Engineering*, vol. 33, no. 4, pp. 361–387, 2009.
[19] "Nwtc design codes (fast by jason jonkman, ph.d.)," http://wind.nrel.gov/designcodes/simulators/fast/. Last modified 05-November-2010; accessed 05-November-2010.
[20] "Nwtc design codes (wt_perf by marshall buhl)," http://wind.nrel.gov/designcodes/simulators/wtperf/. Last modified 17-February-2011; accessed 17-February-2011.
[21] L. Fingersh and K. Johnson, "Controls advanced research turbine (cart) commisioning and baseline data collection," National Renewable Energy Laboratory, Tech. Rep., 2002.
[22] A. Wright, L. Fingersh, and M. Balas, "Testing state-space controls for the controls advanced research turbine," in *Proceedings of the AIAA Aerospace Sciences Meeting and Exhibit*, 2006.
[23] K. A. Stol, W. Zhao, and A. D. Wright, "Individual blade pitch control for the controls advanced research turbine (cart)," *Transactions of the ASME*, vol. 128, pp. 498–505, 2006.
[24] D. Adams, "Model identification and operating load characterization for a small horizontal axes wind turbine rotor using integrated blade sensors," in *Proceedings of the Dynamic Systems and Control Conference*, 2010.
[25] S. Frandsen, "Turbulence and turbulence-generated structural loading in wind turbine clusters," Riso National Laboratory, Tech. Rep., 2007.
[26] "Siemens swt-2.3-101," http://www.energy.siemens.com/mx/pool/hq/power-generation/wind-power/E50001-W310-A121-X-4A00_WS_SWT-2.3-101_US_1009.pdf.
[27] S. Emeis, "Meteorological explanation of wake clouds at horns rev wind farm," *Dewi Maganize*, vol. 37, 2010.
[28] M. Mechali, R. Barthelmie, S. Frandsen, L. Jensen, and P. Rethore, "Wake effects at horns rev and their influence on energy production," in *Proceedings of the European Wind Energy Conference*, 2006.
[29] A. S. Huang, E. Olson, and D. Moore, "Lcm: Lightweight communications and marshalling," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2010.
[30] R. J. Williams, "Simple statistical graident-follwoing algorithms for connectionist reinforcement learning," *Machine Learning*, vol. 8, no. 23, 1992.
[31] J. Peters and S. Schaal, "Policy gradient methods for robotics," in *Proceedings of the International Conference on Intelligent Robots and Systems*, 2006.
[32] J. Tang and P. Abbeel, "On a connection between importance sampling and the likelihood ratio policy gradient," in *Neural Information Processing Systems*, 2011.
[33] S. Boyd and L. Vandenberghe, *Convex Optimization*. Cambridge University Press, 2004.
[34] J. Nocedal and S. Wright, *Numerical Optimization*. Springer, 2006.