Synchrophasing control based on tuned ADRC with applications in turboprop driven aircraft

L. Sheng*, F. Mora Camino,**
X H. Huang***

**College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics,*
JiangSu Province Key Laboratory of Aerospace Power System
Nanjing, China, (Tel: +86-15951935728; e-mail: lsheng@nuaa.edu.cn)
***Ecole Nationale de l'Aviation Civile, Toulouse, 31055 France (e-mail: felix.mora@enac.fr)*
**** College of Energy and Power Engineering, Nanjing University of Aeronautics and Astronautics,*
JiangSu Province Key Laboratory of Aerospace Power System
Nanjing, China, (e-mail: xhhuang@nuaa.edu.cn, correspondence author)

Abstract: Propeller synchrophasing control is an active noise control method which regulates the rotational speed of propellers to maintain their relative phase at an optimal setting to lower noise level. The effectiveness of propeller synchrophasing control is mainly dependent of the rotational speed control. A practical framework for the design and parameter tuning of an Active Disturbance Rejection Controller (ADRC) is proposed in order to reject unknown disturbances. Novel phase command logic is proposed to avoid unreasonable regulation of rotational speed. A filter based on an Extended State Observer (ESO) is used in the presence of measurement noise. Simulation results show that the proposed ADRC controller has a strong ability for disturbance rejection and its new phase control logic makes the system to behave more smoothly with a faster response to perturbation.

Keywords: propeller; noise reduction; synchrophasing control; ADRC; parameter tuning; NSGA_II.

1. INTRODUCTION

Noise and vibrations are serious problems for propeller driven aircraft. Propellers of turboprop engines are working at relative low speed with respect to turbofan and turbojet engines. This is a factor for causing low frequency acoustic noise for humans. According to acoustic theory (Blunt and Rebbechi, 2007; Huang et al., 2014), low frequency noise is more severe for human comfort than high frequency noise. Cabin noise levels for propeller-driven aircraft range from 85 to 100 decibels (dB). For advanced turboprop aircraft a noise reduction of 25 dB is needed to achieve levels comparable to those of turbofan aircraft. In the last two decades an interest has been raised for dealing with this problem. There exist already various methods, which can be classified into two categories i.e. passive control and active control. Traditional passive techniques such as acoustic absorbing material do not have a major effect on noise and vibrations levels of turboprop engines for that propeller vibrations and noise are significant at the low frequencies (50-300 Hertz). In contrast to passive approaches, active noise control approaches have the capability of reducing low frequency noise and vibration over a broadband of frequencies, which is achieved through destructive interference between the primary sound source and the secondary one. Among active noise control strategies, propeller synchrophasing control utilizes the acoustic wave excited by different propellers via adjusting rotational speed to attenuate the noise generated by the other propellers. In this process, no secondary active acoustic source is introduced so this function can be implemented in the engine Full Authority Digital Engine Control (FADEC) system without any additional weight and power. The principle of propeller synchrophasing control can be found in more recent publications by (Huang et al., 2014; Huang et al., 2015) and for the sake of paper length it is not discussed here.

Propeller synchrophasing control is accomplished by slightly adjusting the rotational speed via regulating the pitch angle of the propeller. Obviously the performance is determined by the speed loop of the existing system and can be strongly influenced by the wind turbulence. This turbulence is regarded as an external disturbance while internal disturbances are generated by the vibration of the engines. (Huang et al., 2014; Huang et al., 2015) proved that traditional PID controllers could not be efficient in this case. Then new controller with strong ability to reduce the effects of internal and external disturbances must be introduced. Active Disturbance Rejection Controller (ADRC), proposed by (Han, 2009), seems to be a control approach able to solve such control problems presenting difficulties such as large inertia, strong couplings, time-varying parameters or large internal or external disturbances. The ADRC approach has already been successfully applied to many engineering systems, (Li et al., 2009; Madonski and Herman, 2011; Xia et al., 2014; Przybyła et al., 2012). However, the achieved performance is always dependent of an adequate tuning of the many parameters introduced by ADRC which have some influence on the control performance. Various methods have been already developed to tune these parameters (Gao, 2003; Chen et al., 2011) Among these methods, bandwidth-

parameter tuning by (Gao, 2003) and method non-dominated sorting with genetic algorithm-ii by (Ma et al., 2008) perform rather well, having still some drawback in each of them. So the objective of this paper is to propose a practical framework integrating both methods to tune ADRC design parameters and overcome their limitations to get a better overall performance.

The paper is organized as follows. The ADRC principle is first discussed in section 2. The bandwidth-parameter tuning method and non-dominated sorting with genetic algorithm-ii are introduced and analysed in section 3 and 4 respectively. Then also in section 4, new tuning method is proposed and illustrated through numerical simulation. The implementation of ADRC with propeller synchrophasing control and a novel phase control technic are presented in section 5. Conclusion and final comments are given in section 6.

2. INTRODUCTION TO ADRC

ADRC was at first proposed by (Gao, 2004; Han, 2009), and has been in progress for almost two decades. Now it has been implemented in various applications from different industrial fields. ADRC can be seen as an adaptive controller of conventional PIDs without their important limitations such as: i) inability to eliminate the output error in the presence of inertia; ii) difficulty in using signal derivatives in the presence of disturbances; iii) the use of linear combinations of error signals is less effective than the use of nonlinear ones; iiii) error integration may cause undesired output oscillations and control saturations.

ADRC is a nonlinear control method which originates from sliding mode control (Guo and Jin, 2013). Like PID control, ADRC does not depend on a precise model to produce fast responses with strong robustness. Contrary to PID, ADRC performs in real time the compensation of total disturbances. ADRC is composed of three main functions:

-- A tracking differentiator (TD), which is aimed to enforce a desired transient process for a proper setting;

-- An extended state observer (ESO) which estimates the total disturbance of system;

-- A nonlinear controller (NLC) which turns the system into a pure integrator or a cascade integrator after eliminating unwanted system behavior and disturbances.

The structure of ADRC is show in Fig. 1. The details of these parts are discussed below.

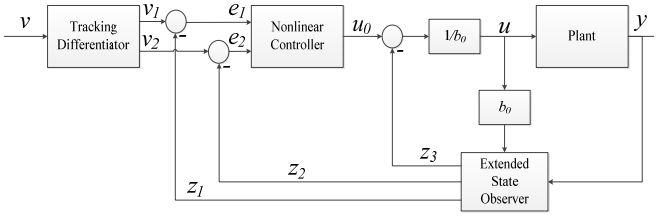

Fig.1. The structure of typical ADRC algorithm.

2.1 Tracking Differentiator

Classic differentiators cannot produce satisfying differential signal because in most occasion stochastic noise in the loop can be augmented which leads to a signal-noise-ratio to an unacceptable level. However, if the differential signal is not calculated from the noisy measured signal, this can be avoided. TD acts as time optimal controller to an ideal but not existed second order system which takes the reference signal as input. Consider a second order cascade integrator system (1)

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u, \quad |u| \le r \end{cases} \tag{1}$$

If x_1 is the tracking of reference signal v in Fig.1, then x_2 can be acquired as the differential signal of x_1 which equals to v. Thus a second order TD can be designed as

$$\begin{cases} \dot{v}_1 = v_2 \\ \dot{v}_2 = fhan(v_1 - v(t), v_2, r, h_0) \end{cases} \tag{2}$$

where $v(t)$ is the reference signal. v_1 is the desired trajectory without overshoot and v_2 is its derivative, r is the speed factor and h_0 is the filtering factor. According to (Han, 1999), a decreasing integration step can lead to limitation of noise but in some cases where the integration step is fixed, one alternative way is to increase the filtering factor to lower noise. Function $fhan(v_1 - v(t), v_2, r, h_0)$ in discrete-time domain is defined by

$$\begin{cases} d = rh_0 \\ d_0 = h_0 d \\ y = (v_1 - v(t)) + hv_2 \\ a_0 = \sqrt{d^2 + 8r|y|} \\ a = \begin{cases} v_2 + \dfrac{(a_0 - d)}{2}\text{sign}(y), & |y| > d_0 \\ v_2 + \dfrac{y}{h}, & |y| \le d_0 \end{cases} \\ \text{fhan} = -\begin{cases} r\,\text{sign}(a), & |a| > d \\ r\dfrac{a}{d}, & |a| \le d \end{cases} \end{cases} \tag{3}$$

2.2 Extended State Observer

Consider a nonlinear second order model of a plant such as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = f(x_1(t), x_2(t)) + d_{ext} + bu \\ y = x_1 \end{cases} \tag{4}$$

where y is the output signal of the plant, u is the plant input signal, d_{ext} denotes the external disturbances. When regarding unknown modeling $f(x_1(t), x_2(t))$ and external disturbances as a total dynamic disturbance, the system from (4) can be rewritten as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 + bu \\ \dot{x}_3 = w(t) \\ y = x_1 \end{cases} \tag{5}$$

where x_3 is the total disturbance which is taken as a state variable. Then the ESO is a Luenberger observer such as:

$$\begin{cases} e_1 = z_1 - y \\ \dot{z}_1 = z_2 - \beta_1 e_1 \\ \dot{z}_2 = z_3 - \beta_2 e_1 + bu \\ \dot{z}_3 = -\beta_3 e_1 \end{cases} \tag{6}$$

where e_1 is the error estimation error for y. Here z_1, z_2 and z_3 are the estimates of the state variables x_1, x_2 and x_3 respectively and β_1, β_2 and β_3 are the observer gains. The controller is chosen such as:

$$u = \frac{u_0 - z_3}{b} \tag{7}$$

Then when estimation has converged, (1) reduces to:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = u_0 \\ y = x_1 \end{cases} \tag{8}$$

which is a second order cascade integrator without any apparent disturbance. In this process, no assumption about the plant is made, in other words, whether its dynamics are linear or nonlinear, time-invariant or variant, with known or unknown parameters, is not relevant. The only condition is that the total disturbance is bounded, which is a reasonable assumption in almost all practical cases.

2.3 Nonlinear Controller

Besides the tracking of the system states and the disturbance-free differential signal, a nonlinear state feedback control law is introduced. It has the following general form:

$$\begin{cases} u_0 = k_p fal(e_1, \alpha_1, \delta) + k_d fal(e_2, \alpha_2, \delta) \\ e_1 = x_1 - z_1 \\ e_2 = x_2 - z_2 \end{cases} \tag{9}$$

where k_p and k_d are proportional and differential coefficients respectively, the α_i are the speed factors, δ is a threshold. All these parameters are taken positive. There *fal* is a nonlinear function aimed to replace the linear combination of error signals to accelerate the convergence of the error signal in a more effective way. An example for *fal* is given here:

$$fal(e, \alpha, \delta) = \begin{cases} \dfrac{e}{\delta^{1-\alpha}}, & |e| \le \delta \\ \text{sign}(e)|e|^{\alpha}, & |e| > \delta \end{cases} \tag{10}$$

and behaves as shown in Fig. 2. It can be seen that for a simple system, the error convergence speed by nonlinear controller is faster than linear where α is set 1.

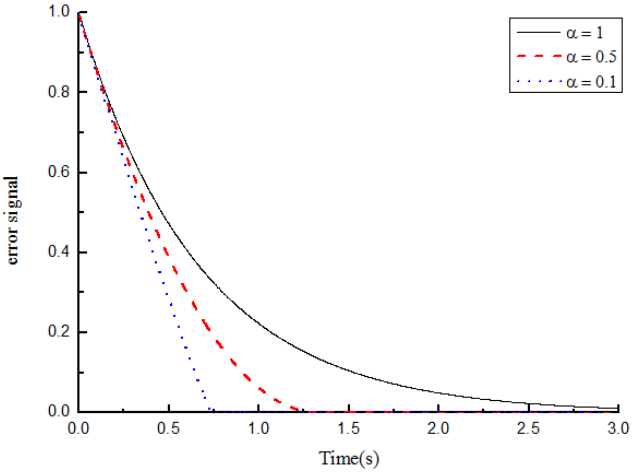

Fig. 2. Evaluation of error signal for different setting of α parameter.

3. BANDWIDTH-PARAMETER TUNING

Although ADRC can estimate the total disturbance and turn the system into a cascade integrator, there are still some parameters which need to be tuned, especially the observer and controller gains. In (Gao, 2003) a bandwidth-parameter tuning method is introduced to cope with the tuning of the observer gains. Rewriting system (5) according to a standard state representation, the system in state space equation is given by:

$$\begin{cases} \dot{x} = Ax + Bu + Dw \\ y = Cx \end{cases} \tag{11}$$

where

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, B = \begin{bmatrix} 0 \\ b \\ 0 \end{bmatrix}, C = \begin{bmatrix} 1 & 0 & 0 \end{bmatrix}, D = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}$$

then the state space observer model is

$$\begin{cases} \dot{z} = Az + Bu + L(y - \hat{y}) \\ \hat{y} = Cz \end{cases} \tag{12}$$

where $L = [\beta_1 \ \ \beta_2 \ \ \beta_3]^{\mathrm{T}}$. Then, combining (11) and (12), the error equation can be written as

$$\dot{e} = A_e e + Dw \tag{13}$$

where

$$A_e = A - LC = \begin{bmatrix} -\beta_1 & 1 & 0 \\ -\beta_2 & 0 & 1 \\ -\beta_3 & 0 & 0 \end{bmatrix}$$

The system will be stable if the roots of the characteristic polynomial of A_e, $\lambda(s) = s^3 + \beta_1 s^2 + \beta_2 s + \beta_3$, are all in the left half complex plane. An easy way to satisfy this first condition is to take the β parameters such as:

$$\lambda(s) = s^3 + \beta_1 s^2 + \beta_2 s + \beta_3 = (s + \omega_0)^3 \tag{14}$$

which needs:

$$\beta_1 = 3\omega_0,\ \ \beta_2 = 3\omega_0^2,\ \ \beta_3 = \omega_0^3 \tag{15}$$

This process is called ω_0-Parameterization where ω_0 is the bandwidth of the state observer. In this way, all the observer poles are placed at - ω_0 in the left half complex plane. So the overall performance of ESO is only dependent on ω_0. (Gao, 2003), to improve the performance of ADRC, proposed to tune simultaneously the controller and observer parameters according to:

$$\begin{aligned} k_p &= \omega_c^2,\ \ k_d = 2\omega_c \\ \omega_0 &= N\omega_c \end{aligned} \tag{16}$$

Then it is supposed that from empirical knowledge, the user can chose factor N between ω_0 and ω_c, where ω_c is the actuator bandwidth.

4. COMBINED TUNING METHOD

Although the parameters could be tuned by considering the above bandwidth-parametrization without using a model of the plant, the chosen value for ω_0 might not be optimal considering the presence of noises and the chosen sampling rate. Hence it appears of interest to introduce a new tuning method.

4.1 Problem Statement

The purpose of tuning methods for the parameters of ESO is to make the output of the observer model as close as possible to the real trajectory of the system states. For a working point, a linear continuous system model can be obtained and transformed into the discrete time domain using the Z-transform. Then in this discrete time domain, two cost functions assessing respectively accuracy and fastness of tracking are adopted as the criteria for the tuning method:

$$\begin{cases} J_1 = \sum_{t=t_0}^{t_f} \left[(x_1^t - z_1^t)^2 + (x_2^t - z_2^t)^2 + (x_3^t - z_3^t)^2 \right] \\ J_2 = \max\{\tau_1,\ \tau_2,\ \tau_3\} \end{cases} \tag{17}$$

where x and z are system states and observer states respectively, τ_i is the settling time of state variable i. In (14) t_0 is the start time of an external impulse used to activate the system, t_F is chosen to cover the whole ESO reaction to this impulse. The τ_i is defined by:

$$\tau_i = \arg\min(t) \ \text{ with } \ x_i^{t+1} = x_i^t$$

Once estimator parameters are chosen so that the two cost functions are simultaneously minimized, the ESO will achieve a good performance. Therefore, an algorithm to achieve multi-criteria optimization is needed.

4.2 Non-dominated Sorting Genetic Algorithm-ii

As there is no analytical relation between the tuned parameters and the chosen performance indexes, classical mathematical programming approaches to explore the non-dominated solutions (Han, 2009) cannot be applied here. So Genetic Algorithms (Dias and Vasconcelos, 2002) coupled with a discrete time domain model of the process appear to be an effective solution approach. Among the large variety of genetic algorithms available today, it appears that the Non-dominated Sorting Genetic Algorithm-ii (NSGA_II) which was first proposed by (Deb et al., 2002) allows coping with multi-criteria optimization problems. Figure 3 displays the different steps of this algorithm.

For evaluation purpose of each individual of the current population P, the expression retained for the continuous transfer function of the pitch angle to the propeller rotational speed at the chosen working point is:

$$G(s) = \frac{-0.5213s - 2.435}{s^2 + 13.23s + 39.61} \tag{18}$$

Discretizing the continuous-time dynamic system model (18) using zero-order hold on the inputs and a sample time of 0.02 seconds in Matlab, the associate discrete transfer function is given by:

$$G(z) = \frac{-0.9583z + 0.8728}{z^2 - 1.754z + 0.7675} \tag{19}$$

This leads to the recursive expression of the system:

$$y(k) = 1.75y(k-1) - 0.77y(k-2) - 0.96u(k-1) + 0.87u(k-2) \tag{20}$$

where $y(k)$ is the output and $u(k)$ is the control variable at k-th step in discrete domain.

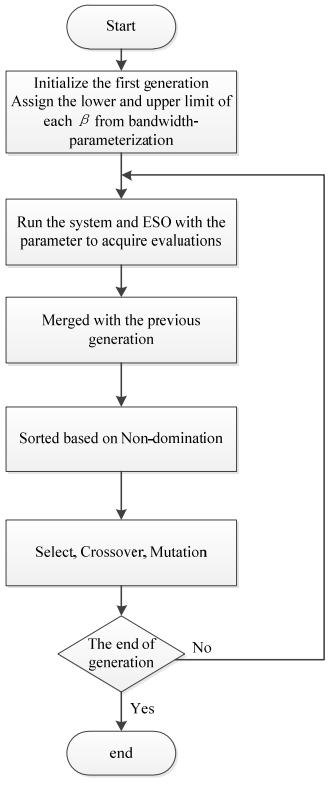

Fig. 3 Flow charts of NSGA_II.

The sorting process of NSGA_II will generate dominating sets among solutions (the individuals) by comparing their computed performances (J_1 and J_2). The different steps of the adopted sorting algorithm are the following where F_l is the non-dominated set; Q is the non-dominated subset of each individual in F_l:

Step1. For each individual p belonging to main population P, for each individual l belonging to P, initialize $S_p = \emptyset$, $F_l = \emptyset$, $n_p = 0$;

① if p dominates l, $S_p = S_p \cup \{l\}$;
if l dominates p, $n_p = n_p + 1$;

② if $n_p = 0$, $p_{\text{rank}} = 1$, $F_1 = F_1 \cup \{p\}$;

Step2. Set $i = 1$;

Step3. For $F_i = \emptyset$, $Q = 0$;

① for each q belonging to F_i, for each l belonging to S_p, let $n_l = n_l - 1$, if $n_l = 0$, $q_{\text{rank}} = p_{\text{rank}} + 1$, $Q = Q \cup \{l\}$;

② $i = i + 1$;

③ $F_i = Q$;

Step4. For each F_i, n is the number of individuals in F_i.

① initialize crowded distance $i(d_k) = 0$, k = 1, 2, 3, …, n;

② for each cost function J_m
Sort the individuals based on cost function J_m
Define an infinite distance to the boundary individual, which means $i(d_1) = i(d_n) = \infty$
For k varying from 2 to n-1

$$i(d_k) = i(d_k) + \frac{J_m(k+1) - J_m(k-1)}{J_m^{\max} - J_m^{\min}}$$

where J_m is the m-th cost function of the k-th individual;

Step5. Assume that every individual i has two attributes which it can be selected through:

① non-domination rank (i_{rank});

② crowded distance $i(d)$;

A partial order $\prec$ is defined by:

$$\begin{aligned} & i \prec j \quad \text{if} (i_{rank} < j_{rank}) \\ & \text{or} \ (i_{rank} = j_{rank}) \\ & \text{and} \ (i(d) > j(d)) \end{aligned}$$

That is, between two individuals with differing non-domination ranks, the one with the lower (better) rank is preferred. Otherwise, if both individuals belong to the same front, the individual that is located in a lesser crowded region is preferred. More details about non-domination rank and crowded distance can be found in (Deb et al., 2002).

In the crossover process, the possibility of the i-th individual crossover is calculated as follows:

$$\pi_i = \begin{cases} \pi_{\max}, & J_i < \sum_i J_i \Big/ |P| \\ \pi_{\min} + \dfrac{\sum_i J_i \Big/ |P| - J_i}{\pi_{\max} - \sum_i J_i \Big/ |P|} (\pi_{\max} - \pi_{\min}), & J_i \geq \sum_i J_i \Big/ |P| \end{cases} \tag{21}$$

where $\pi_{\max}$ and $\pi_{\min}$ are the maximum and minimum possibility respectively, J_i is the evaluation of the i-th individual and $|P|$ is the number of individuals. Similarly, the possibility of the i-th individual in the mutation process is given as

$$\pi_i = \begin{cases} \pi_{\min}, & J_i < \sum_i J_i \Big/ |P| \\ \pi_{\min} + \dfrac{J_i - \sum_i J_i \Big/ |P|}{\pi_{\max} - \sum_i J_i \Big/ |P|} (\pi_{\max} - \pi_{\min}), & J_i \geq \sum_i J_i \Big/ |P| \end{cases} \tag{22}$$

where the definition of the parameters is the same as (21).

4.3 Simulation Result

By combining the bandwidth-parameterization and the NSGA_II, a global approach of ADRC parameters tuning is proposed, where the implicit range assignment of two bandwidths is avoided and without a trail-and-error procedure, more accurate values for the parameters are determined.

Then, setting ω_c to be 4, the settling time is tuned to be equal to one second. Then define the range $\omega_0 = 1 \sim 10\omega_c$. Thanks to the separation principle, ESO could be tuned with a fixed control law. After running the NSGA_II with initial setting of population equal to 80 and a generation of 100, the ESO parameters were found such as $\beta_1 = 50.5$, $\beta_2 = 120.3$, $\beta_3 = 450.7$ and these parameters are defined as parameters set 1.

The comparisons are made between the simulation results using parameters set 1 above and set 2 where $\beta_1 = 100$, $\beta_2 = 200$, $\beta_3 = 600$. Through Fig.4 to Fig.6 good agreement between the real system and tuned ESO set 1 can be seen and there is slight mismatch between the system states and not tuned ESO set 2, especially for the augmented state x_3. The agreement made by parameters set 2 is worse than set 1 for parameters in set 2 are chosen arbitrarily and not tuned. Comparation is also made between set 1 and 2 during the engine start up. The results are illustrated in Fig.7 and Fig.8.

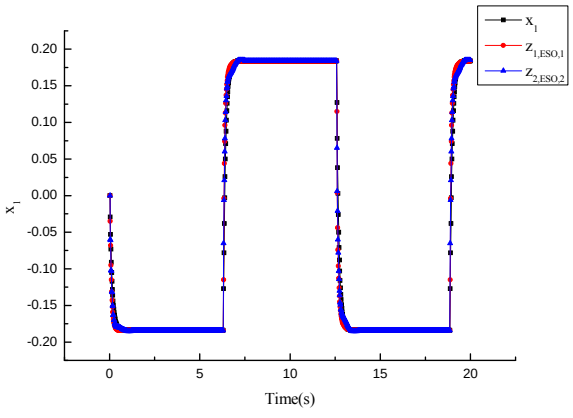

Fig. 4. Comparison of tracking of the system state x_1.

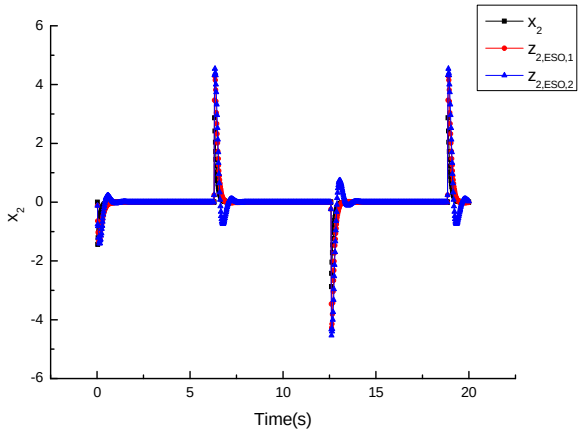

Fig. 5. Tracking of the system state x_2.

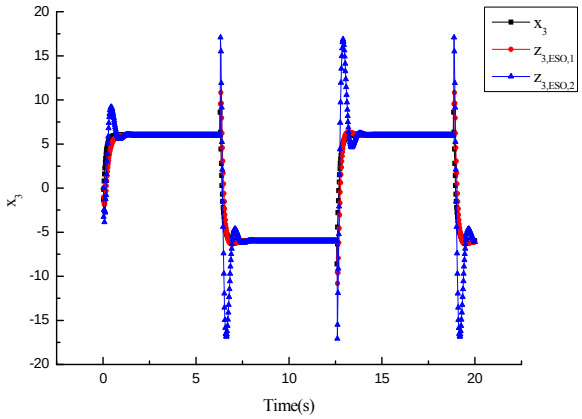

Fig. 6. Tracking of the total disturbance x_3.

As can be seen in Fig. 7 and Fig. 8 by tuned parameters, the agreement is superior to that of arbitrary parameters set.

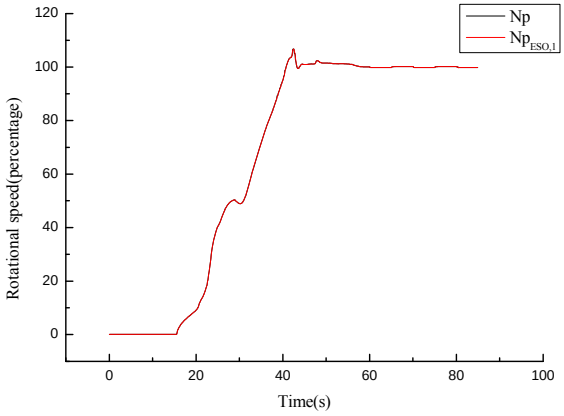

Fig. 7. State tracking of rotational speed (N_p) during engine start up process using ESO parameters set 1.

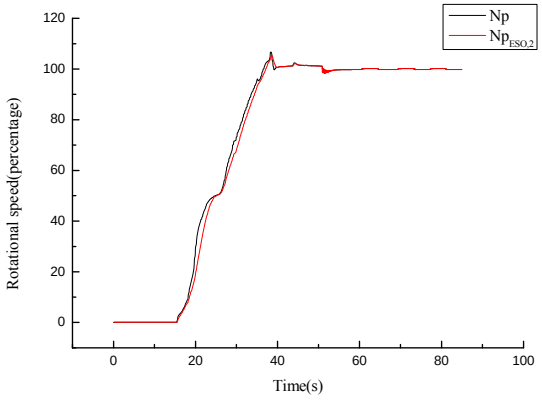

Fig. 8. State tracking of rotational speed (Np) during engine start up process using ESO parameters set 2.

5. CONTROL STRATEGY

Before introducing the control strategy, the structure of the system is illustrated below. The existing engine system has two independent inputs i.e., fuel flow and propeller pitch angle. In a previous work, (Huang et al., 2015) has demonstrated that a limit cycle implying oscillations will show up if synchrophasing control is integrated it in the rotational speed loop. Speed/power command correction is an alternative and the basic block diagram with tuned ADRC controller is shown in Fig. 9.

There α_{ref} is the phase command signal, $\alpha_{1,a}$ and $\alpha_{2,a}$ are the actual phases which are the integration of the propeller speeds of engine 1 and 2 respectively. P_{ref} and P_{out} represent the command and measurement signals of engine power while Np denotes the rotational speed of the propeller. The control variables of the two engines are the fuel flow w_f and the pitch angle β. One way to reduce the nonsynchronous effects is to introduce a virtual engine operated at constant speed, which is also the speed command, without disturbance in software level and assign the phase of the virtual engine to be zero at all times. Thus, the phase for each engine is calculated by integrating the difference between the speed command and the speed measurement. This virtual engine setting can be easily extended to the case where the number of engines is more than 2.

New phase command logic is applied in this case. Although there is no priority in any engines, to be simple, let engine 1 to be the ideal master engine and engine 2 to be the ideal slave engine. Also let engine 1 track the phase of the virtual engine, that is $\alpha_{1,\,ref} = 0$, so α_{ref} is actually the command of engine 2 which means $\alpha_{ref} = \alpha_{2,\,ref}$. Consider a propeller configured with 6 blades, the phase of this engine is within -30° ~ +30°. If now the operating phase is 25° while next command is -10°, current logic is to decelerate propeller for a while to achieve that phase. But the displacement of phase is 35° which is not the optimal one. Is there a better path? If the propeller is accelerated, the phase will reach 30° and above 30° it will become negative until it is settled to -10°. It can be seen in this routine that the displacement of phase is 25° which is superior to 35°. The reason is that according to

(Niessen, 1991) the phase is the monotonic integration of the speed difference in the range of -30° ~ 30° and positive ones which stand for the integrations over the whole time are positive, vice versus the negative. So if the displacement command is negative and larger than 30°, the integration should be reduced and the deviation should be smooth without additional constraint. A chart here is presented in Fig. 10 to illustrate the new logic.

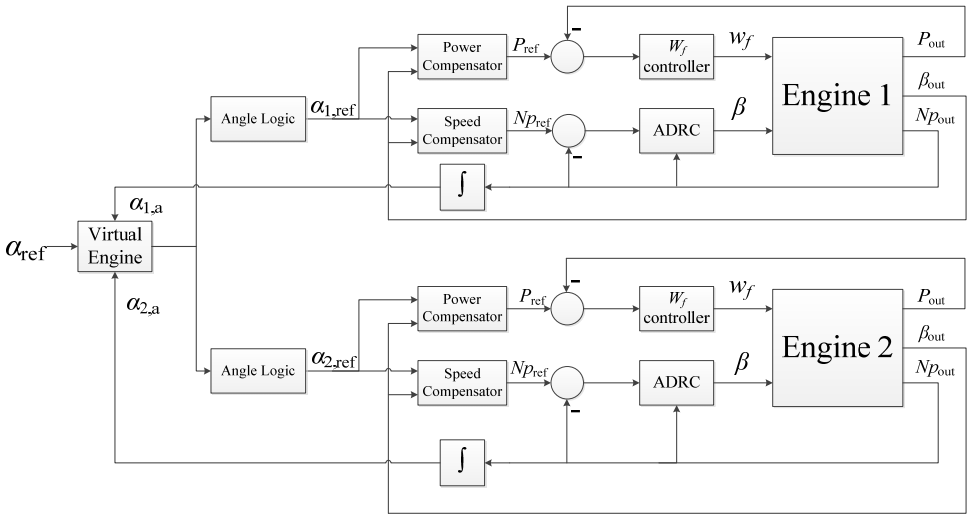

Fig. 9. Block diagram of the integrated synchrophasing control.

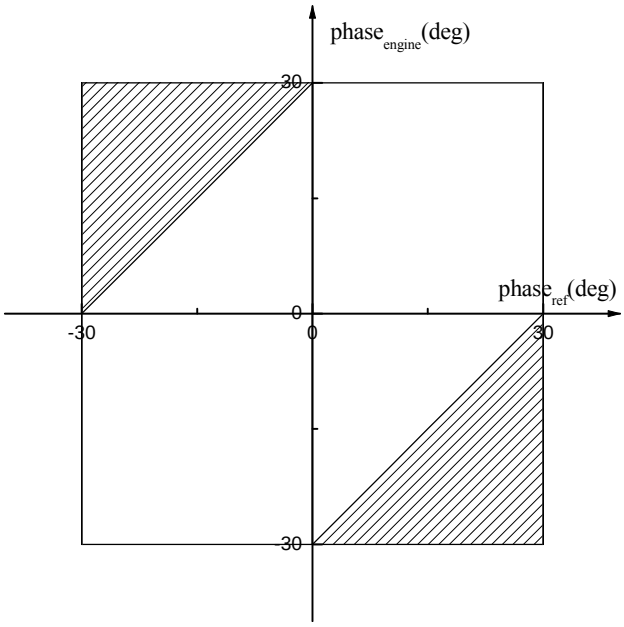

Fig. 10. Corrected Phase Logic.

If the phases of the engine and the command fall into the non-shaded area, then the traditional logic is adopted while the new logic is adopted if the phases fall into the shaded regions where $|\alpha_{engine} - \alpha_{ref}| > 30°$. Here 30° is the result of equation $\alpha = 180°/n$, where n is the number of blades. This logic could be extended easily to the case where the number of engines is more than two.

The tuned ADRC and the new logic for phase command lead to the simulation shown in Fig.11and Fig.12.

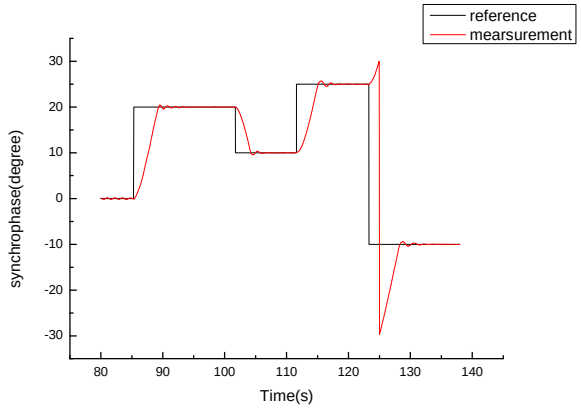

Fig. 11. Synchrophase responses without disturbance

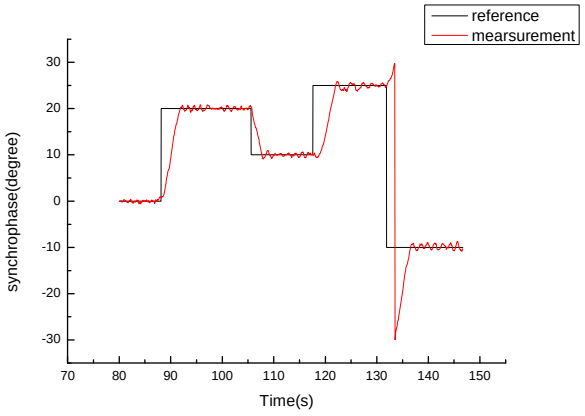

Fig. 12. Synchrophase responses with disturbance.

The simulation adopted a white Gaussian noise with a standard deviation of 0.1 to represent the rotational speed disturbance.

Filters are often used in the presence of measurement noise in the system. In this paper, a simple inertial filter is adopted in the ESO modulus to reduce the influence of measurement noise. Then ESO (6) can be rewritten as

$$\begin{cases} \tilde{y} = \tilde{y} - \alpha_{filter}(\tilde{y} - y) \\ e_1 = z_1 - \tilde{y} \\ \dot{z}_1 = z_2 - \beta_1 e_1 \\ \dot{z}_2 = z_3 - \beta_2 e_1 + bu \\ \dot{z}_3 = -\beta_3 e_1 \end{cases} \tag{23}$$

where $\tilde{y}$ is the output of the inertial filter and α_{filter} is the filter gain. Three different values i.e., 0.1, 0.5 and 0.9 for filter gain have been used in the simulation. The results are illustrated in Fig. 13 to Fig.16.

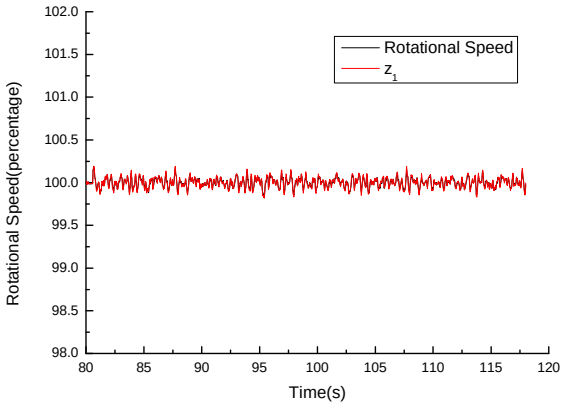

Fig. 13. ADRC without filter.

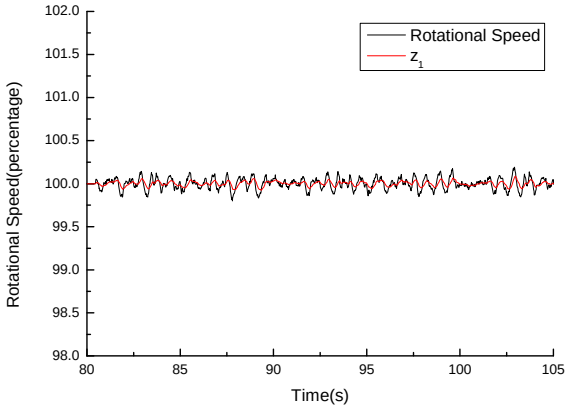

Fig. 14. ADRC with filter factor setting at 0.1.

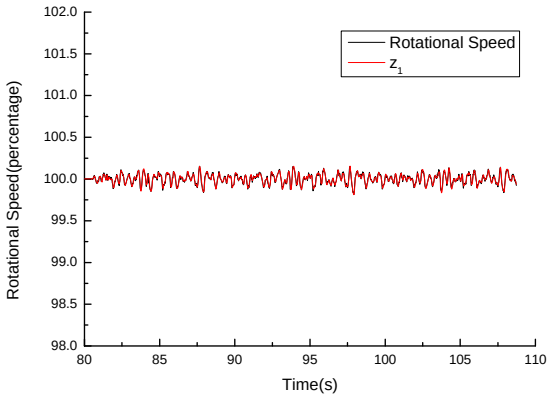

Fig. 15. ADRC with filter factor setting at 0.5.

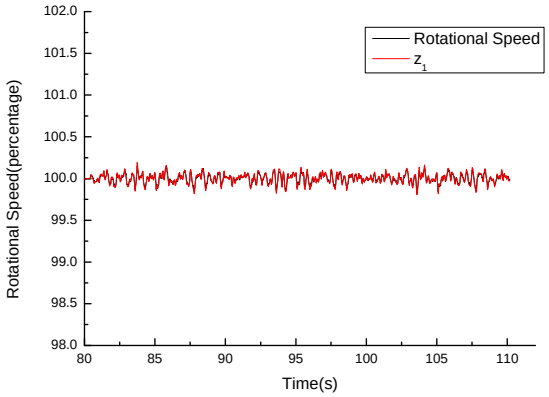

Fig. 16. ADRC with filter factor setting as 0.9.

It can be seen that larger the filter is, the less effect can be achieved while the delay between the state of the system and the ESO tracking is less. This is because the introduction of the inertial filter will augment the phase delay of the system, which is undesirable in this case.

5. ACKNOWLEGDGEMENT

This work is supported by the National Natural Science Foundation of China (No. 51576097) and Jiangsu Innovation Program for Graduate Education (grant no: KYLX15_0253) .

6. CONCLUSIONS

In this paper, a practical ADRC tuning methodology is presented based on bandwidth-parametrization and non-dominated sorting generic algorithm-II. This routine makes tuning more reasonable with the consideration of the constraint applied on the system, i.e. sampling rate. A new phase command logic is also presented in this paper. Meanwhile, ADRC technique is adopted in synchrophasing control and it shows the great potential in rejecting disturbance.

REFERENCES

Blunt, D. and Rebbechi, B. (2007). Propeller Synchrophase Angle Optimisation Study. *13th AIAA/CEAS Aeroacoustics Conference (28th AIAA Aeroacoustics Conference)*.

Chen, X., Li, D., Gao, Z. and Wang, C. (2011). Tuning Method for Second-order Active Disturbance Rejection Control. In: *Proceedings of the 30th Chinese Control Conference*. Yantai.

Dias, A. and de Vasconcelos, J. (2002). Multiobjective genetic algorithms applied to solve optimization problems. *IEEE Transactions on Magnetics*, 38(2), pp.1133-1136.

Deb, K., Pratap, A., Agarwal, S. and Meyarivan, T. (2002). A fast and elitist multiobjective genetic algorithm: NSGA-II. *IEEE Transactions on Evolutionary Computation*, 6(2), pp.182-197.

Gao Z., (2003). Scaling and bandwidth-parameterization based controller tuning. *Proceedings of the 2003 American Control Conference, 2003.*

Gao Z., (2004). On discrete time optimal control: A closed-form solution. *American Control Conference, 2004. Proceedings of the 2004*, pp. 52-58.

Guo, B. and Jin, F. (2013). Sliding Mode and Active Disturbance Rejection Control to Stabilization of One-Dimensional Anti-Stable Wave Equations Subject to Disturbance in Boundary Input. *IEEE Transactions on Automatic Control*, 58(5), pp.1269-1274.

Han, J. (2009). From PID to Active Disturbance Rejection Control. *IEEE Transactions on Industrial Electronics*, 56(3), pp.900-906.

Han, J. and Yuan, L. (1999). The Discrete Form of the Tracking Differentiator. *Systems Science and Mathematical Sciences*, 19,pp.268-273.

Huang, X., Sheng, L. and Wang, Y. (2014). Propeller Synchrophase Angle Optimization of Turboprop-Driven Aircraft—An Experimental Investigation. *Journal of Engineering for Gas Turbines and Power*, 136(11), p.112606.

Huang, C., Wang, Y. and Sheng, L. (2015). Synchrophasing control in a multi-propeller driven aircraft. *2015 American Control Conference*.

Li, S., Yang, X. and Yang, D. (2009). Active disturbance rejection control for high pointing accuracy and rotation speed. *Automatica*, 45(8), pp.1854-1860.

Ma Q., Xu D., and Shi Y., (2008). Research of synthesis tuning algorithm of Active-Disturbance-Rejection Controller. *2008 7th World Congress on Intelligent Control and Automation*.

Madonski, R. and Herman, P. (2011). An experimental verification of ADRC robustness on a cross-coupled Aerodynamical System. *2011 IEEE International Symposium on Industrial Electronics*.

Niessen, F. (1991). *Propeller synchrophaser control with trajectory logic*. 5042965.

Przybyła, M., Kordasz, M., Madoński, R., Herman, P. and Sauer, P. (2012). Active Disturbance Rejection Control of a 2DOF manipulator with significant modeling uncertainty. *Bulletin of the Polish Academy of Sciences: Technical Sciences*, 60(3).

Xia, Y., Dai, L., Fu, M., Li, C. and Wang, C. (2014). Application of active disturbance rejection control in tank gun control system. *Journal of the Franklin Institute*, 351(4), pp.2299-2314.