Modeling and Simulation of a Single Gain Tuning ADRC Controller in Matlab/Simulink

Mohammad Kia
Control Department
Datis Elevator
Tehran, Iran
m_kia@hotmail.com

Pouya Mansouri
Control Department
Datis Elevator
Tehran, Iran
mansouri@datis-elevator.ir

Ali Javdani
Control Department
Datis Elevator
Tehran, Iran
javdani@datis-elevator.ir

***Abstract*—Active Disturbance Rejection Control (ADRC) is one of the recently proposed and best approaches to control diverse range of plants. However, optimized parametrization of the controller along with a proper Matlab/Simulink simulation has been an issue so far. In this research, after studying the basic principle and structure of ADRC, we have discussed how the controller parameters can be scaled so that the tuning parameters can be reduced to only one. Based on the studies, a user defined ADRC block library was created in Matlab/Simulink by developing m-functions for special linear and nonlinear functions and masking subsystem for building user custom library. The examples show that the ADRC simulation model can be used in graphical modeling and the block parameters are easy to modify. Also, the simulation and practical results show that the modeling method is valid and practical. Furthermore, the proposed library is easy to extend and related suggestions are made.**

Keywords—ADRC, Single-Gain-Tuning, Feedforward, Input Profile, Matlab/Simulink/Simscape, Masking Subsystem

I. INTRODUCTION

Recent developments in the field of control systems, especially in digital controllers utilized in embedded-systems, has drawn the designers' attention to ADRC structure. The structure of the controller is such that it can be easily implemented in any programming language, and its computational complexity is so low that it can be executed in almost any embedded platform for real-time purposes. Because of these benefits and advantages in addition to the robustness of the controller, the applications of ADRC is increasing day by day [1-5]. These applications include motion controls, motors and vector controls, power inverters, hydraulics, chemical processes, temperature control and so on.

ADRC was first introduced by J. Han [6] in his paper and almost all the principles and structure are described clearly. But, there are two important issues to consider in the proposed structure in [6]: 1) the number of the tuning parameters 2) the input profile and the feedforward compensation within.

As for the first issue, in most of the recent researches, ADRC structure have been introduced as a complex controller with many parameters to design and may tune. These parameters include the Extended State Observer (ESO) gains, Nonlinear State Error Feedback (NLSEF) gains and the input profile known as the Tracking Differentiator (TD). Some works like the researches performed in [7-10] have focused on relating these parameters together, and among them [11] have proposed an outstanding relation between the control bandwidths and parameters. The work done in [11] is the one on which we mainly base our mathematics. However, it does not consider the second issue, input profile and feedforward compensation.

There is no doubt that in order to completely remove tracking delays, the controller must help the plant to act by the means of the feedforward control signal, before the tracking feedback error happens and cause the control signal to change. More information on this matter can be found in [12]. The control structure proposed in [6] somehow applies the feedforward as the second input of the NLSEF, but still it is not enough.

The reader should consider the fact that every controller tries to make the system follow the reference signal as exactly as it can. If the input is just a step function, system output will try to act as a step. For example, if the plant is the position control of a high inertia mechanical system, the perfect controller with a step input will result in the fracture of all the joints and the gearbox at once, because no mechanical system can bear the change of position in a step pattern. This gets worse when the controller includes differentiators. They produce impulses in the control signal which will be fed to the plant and results in unwanted disturbances. This can also happen even if the reference signal is a ramp; the controller of a second order system can have second order differentiator, so the result will be also disastrous. So we should always give the step as a reference plan to a profile/trajectory generator and use the generated profile/trajectory as the input of the controller.

Since the ADRC is for the systems with cascade integrations, the derivatives of the input profile can be used as feedforward for the controller. Reference [12] illustrates this matter clearly.

The main goal of this research is to propose an effectual ADRC block library for Simulink. Simulink modeling based on the structure in [6] have been presented in [13-15], but we propose the improved library block based on the optimized ADRC structure.

In the following sections, we first define the optimized structure, afterwards we design the Simulink library block, following that we test the library in two case studies, and finally we validate the theories and simulations by practical results.

This work was supported in full by Mehbang Industrial Group
http://mehbang.group/en

II. ADRC Mathematical Model

A. Controller Structure

ADRC is in fact a robust and effective controller, but every part of the controller needs to be fully understood so that the implementation could be done as simple and effective as it can be in practice. ADRC is generally composed of three parts/modules:

1. The input profile which is mostly known as TD.
2. A linear/nonlinear full state feedback controller which is introduced as NLSEF.
3. A linear/nonlinear extended state observer which has more orders than the traditional Luenberger observer and is referred to as ESO.

Fig.1 shows the overall structure of the controller used in this work. The profile generator creates the reference signal as well as its derivatives that will be used to help the controller to also converge the error derivatives to zero. Depending the order of the plant, different types of trajectories may be used including second or third order time-optimal, minimum-jerk, minimum-snap and so on. Sufficient information on trajectory generators can be found in [16-19]. The structure also includes the correct location of the control signal saturator. Mathematics on the NLSEF and ESO modules are described as following.

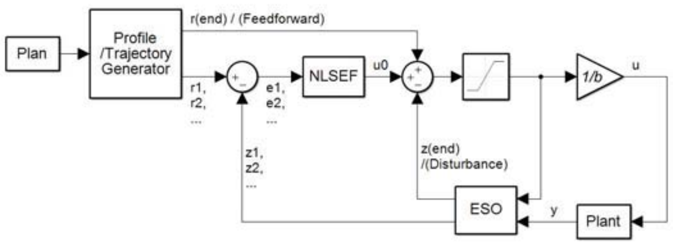

Fig. 1. Optimized ADRC Topology

Consider the following state-space of a general linear system when the observability and controllability conditions hold (pair matrices A, C are observable):

$$\begin{cases} \dot{X} = AX + BU \\ Y = CX + DU \end{cases} \tag{1}$$

A general *nth* order plant with unknown dynamics and external disturbances suitable for ADRC will be in the form of:

$$y^n = f\left(t, y, \dot{y}, \dots, y^{(n-1)}, u, \dot{u}, \dots, u^{n-1}, w\right) + bu \tag{2}$$

B. Extended State Observer Design

Han's observer can be derived starting from the extended state equation:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_n = x_{n+1} + bu \\ \dot{x}_{n+1} = h \\ y = x_1 \end{cases} \tag{3}$$

with $x_{n+1} = f$ added as an augmented state, and the unknown disturbance derivative $h = \dot{f}$. For the state equation defined in (1) and (3) we can write the so-called Luenberger observer as:

$$\begin{cases} \dot{\hat{x}} = A\hat{x} + Bu + L(y - \hat{y}) \\ \hat{y} = C\hat{x} + Du \end{cases} \tag{4}$$

Where the state matrices are defined as:

$$A = \begin{bmatrix} 0 & I_{n\times n} \\ 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0_{(n-1)\times 1} \\ b \\ 0 \end{bmatrix},\ C = [1 \quad 0_{1\times n}],\ D = 0,$$
$$L = [l_1 \quad l_2 \quad \dots \quad l_{n+1}] \tag{5}$$

The observer error $e = x - \hat{x}$ satisfies the equation:

$$\dot{e} = (A - LC)e \tag{6}$$

In this work, the eigenvalues of *A-LC* are made Hurwitz and put all at ω_o so that the observer error can converge to zero and the observer parameter be easy to tune [11]. Having a single output, we define $e = y(t) - z_1$ as the observer error. Finally the ESO will have the form of:

$$\begin{cases} \dot{z}_1 = z_2 + l_1 e \\ \dot{z}_2 = z_3 + l_2 e \\ \dots \\ \dot{z}_n = z_{n+1} + bu + l_n e \\ \dot{z}_{n+1} = l_{n+1} e \end{cases} \tag{7}$$

In this observer we will have $z_1, z_2, \dots, z_n$ converged to $y, \dot{y}, \dots, y^{(n-1)}$ and $z_{n+1} \to f$ as the disturbance term.

C. Full State Feedback Controller Design

By choosing the control law as $u = \frac{u_0 - z_{n+1}}{b}$ we will have the system in (2) simplified as $y^{(n)} = (f - z_{n+1}) + u_0 \approx u_0$ which reduces the plant to approximately a unit gain cascaded integrator plant and (3) can be written as:

$$\begin{cases} \dot{x}_1 = x_2 \\ \dot{x}_2 = x_3 \\ \dots \\ \dot{x}_n = u_0 \\ y = x_1 \end{cases} \tag{8}$$

In the simplified state equation we can easily choose the control law $u_0 = -Kx$ as the full state feedback controller. Substituting into the state space equations form (1), we have:

$$\dot{x} = (A - BK)x \tag{9}$$

Where the state matrices are modified as:

$$A = \begin{bmatrix} 0 & I_{(n-1)\times(n-1)} \\ 0 & 0 \end{bmatrix}, \qquad B = \begin{bmatrix} 0_{(n-1)\times 1} \\ b \end{bmatrix}$$
$$K = [k1 \quad k2 \quad \dots \quad kn] \tag{10}$$

Once more, the eigenvalues of *A-BK* are made Hurwitz and put all at ω_c so controller error can converge to zero and the control parameter be easy to tune [11]. Having the profile generator produce all the orders of the input profile, we can write the control law u_0 as:

$$u_0 = k_1e_1 + k_2e_2 + \cdots + k_ne_n + r_{n+1} \\ e_i = r_i - z_i \tag{11}$$

Where r_i are the reference signal and its derivatives and r_{n+1} is the $(n+1)^{th}$ order of the reference signal which is used as the feedforward controller signal.

D. Solution to the Gain Values

Solving the two equations $eig(A - BK) = w_c$ and $eig(A - LC) = w_o$ will result in the Hurwitz polynomial as follows:

$$\begin{cases} (S+\omega_c)^n = S^n + k_nS^{n-1} + \cdots + k_2S + k_1 \\ (S+\omega_o)^{n+1} = S^{n+1} + l_1S^n + \cdots + l_nS + l_{n+1} \end{cases} \tag{12}$$

An important note here is the relation between ω_c and ω_o. We define this relation as $\omega_c = R \times \omega_o$ and R is the defined the bandwidth ratio. In most cases we can consider R=1. But in some cases in which we need more agility, especially in real-time applications, R is considered between 0 and 1.

E. Adding Nonlinearity

The non-weighted linear sum used in the PID controllers and the full state feedback is a traditional way to mix all the control signals, but there can be other potential combinations which can be much more effective. As an alternative, Han in [6] has presented the so-called *fal* function. In the error signals discussed in (7) and (11), the controller errors (e_i) and the observer error (e) can both be replaced by *fal(e_n)* in which *fal* is defined as:

$$fal(e,\alpha,\delta) = \begin{cases} \frac{e}{\delta^{1-\alpha}}, & |x| \le \delta \\ |e|^\alpha sign(e), & |x| \ge \delta \end{cases} \tag{13}$$

Note: In many applications, α parameter varies by the order of the error signal. In embedded applications, it is easier to use the powers of two fractions since all processors support SQRT function. For example, 1 is used for e_1, 0.5 (=$2^{1/2}$) is used for e_2, 0.25 (=$2^{1/4}$) is used for e_3 and so on.

III. Simulink Modeling of The Controller

Simulink modeling of the controller in the presented structure and parametrization format is quite handy and easy. First a subsystem must be created, then these blocks must be added to it: two Matlab Functions, one Discrete-Time Integrator, one Saturation, two Input and one Output. Fig.2 shows a sample subsystem created in Simulink. A Mask is created on the designed subsystem with the Dialog Box Parameters presented in Table.1. The upper and lower limits of the saturation block are accordingly set to "sat*b" and "-sat*b" values. Since ADRC is a basically a digital controller, a discrete-time integrator is used, and its initial value is set to "zeros (1, n + 1)". From a geometric perception of the ADRC topology presented by Han in [6], it can be easily deduced that the system input gain b can be separated from the controller parts namely NLSEF and ESO. Thus, we have put it at the output of the controller as a simple forward gain and completely omitted from the calculations performed in NLSEF and ESO.

TABLE I. Dialog Box Parameters

Field Number	Parameter	Variable
#1	Input Gain (b)	b
#2	Bandwidth (w)	w
#3	System Order	n
#4	Output Saturation	sat

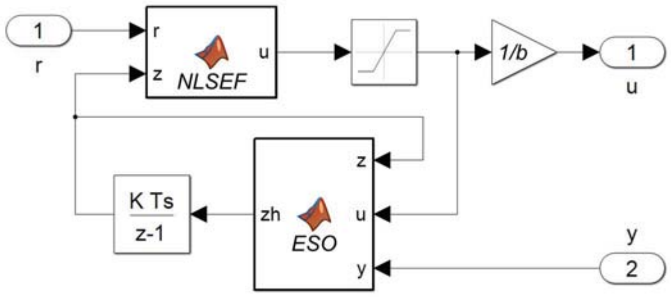

Fig. 2. ADRC subsystem

In order to create A, B, L and K matrices for calculations in NLSEF and ESO, an initialization code has been written in the "Initialization commands" section of the created Mask:

```
A = zeros(n + 1);
A(1:end-1, 2:end) = eye(n);
B = zeros(n, 1);
B(end) = 1;
B = [B; 0];
syms s;
eqn = (s + w) ^ n;
c = fliplr(sym2poly(eqn));
K = c(1:end-1);
eqn = (s + w) ^ (n + 1);
c = sym2poly(eqn);
L = c(2:end)';
```

The code is based on finding the solution of equation (12) with the default R=1 as bandwidth ratio. Since simplicity is the first priority in this research, all the computations in the block are based on state space matrix calculations.

The remaining things in the block design are the Matlab codes for NLSEF and ESO functions which are following:

```
function u = NLSEF(r, z, K)
    u0 = K * (r(1:end-1) - z(1:end-1));
    u = u0 + r(end) - z(end);
end

function zh = ESO(z, u, y, A, B, L)
    zh = A * z + B * u + L * (y - z(1));
end
```

Note: The "Data Scope" of the input variables "r, z, y, y" is set to "Input" and for the input variables "K, A, B, L" is set to "Parameter" which will be set from the global initializations performed in "Initialization commands".

From Table.1 it is clear that the controller block has 4 tuning parameters. The output saturation needs no comments, and the Controller Bandwidth is the only tuning parameter of the controller that the operator needs for tuning process, which mathematically is $\omega = \omega_c = \omega_o$ according to (12).

As for the system order and input gain b, the reader should note that a good controller is always the one designed for a properly modeled and identified system. The properness here does not mean that one should model every bit of details of the target system. It merely means that the designer should properly estimate the order of the system and the input gain b. These parameters are not to be handed to the "Operator in Filed" for tuning, in a practical use. The controller designer can easily estimate the order of the plant, but for parameter b, it is highly suggested that an automatic identification process be designed to calculate the value of b for every system. For example, in the mechanical plant example previously mentioned, the b parameter is the inverse of the system inertia. Automated inertia identification algorithms can be found in [20, 21]; these works offer a very good vision on how one can design a b identification algorithm. An online b estimator gives the controller adaptation.

If the reader needs to add nonlinearity parts to the design, it is simply possible by adding the following functions to NLSEF or ESO. The example below is for 3rd order ESO, but the same approach can be used for NLSEF and other orders.

```
function zh = ESO(z, u, y, A, B, L)
    zh = A * z + B * u + L .* fe(y - z(1));
end

function x = fe(e)
    x = zeros(3, 1);
    d = 0.001;
    x(1) = fal(e, 1.00, d);
    x(2) = fal(e, 0.50, d);
    x(3) = fal(e, 0.25, d);
end

function e = fal(x, a, d)
    if abs(x) <= d
        e = x / d ^ (1 - a);
    else
        e = abs(x) ^ a * sign(x);
    end
end
```

IV. Simulations

In this section we intend to validate what discussed earlier by running some simulations. The selected plant is a simple mechanical rotational plant including an inertia, a rotational damper acting as a viscose friction and a first order low-pass filter to simulate time delays such as motor dynamics and the electric drive. The dynamic equation and transfer function of the inertia and damper system (which is the common model for motion control platforms) is as following:

$$\begin{cases} J\dot{\omega} = \tau - B\omega \\ \Rightarrow G_P = \frac{y}{u} = \frac{1}{JS+B} \end{cases} \tag{14}$$

Where J is plant inertia, B is viscose friction (damping) coefficient, τ the input torque and ω the output rotational velocity. Therefore, the system order is 1 and the input gain b is the inverse of plant inertia. The mechanical modeling is performed in Simscape, which is a perfect tool for modeling mechatronic systems. Fig.3 shows the overall Simulink model. The sample time used in the simulations is 1ms.

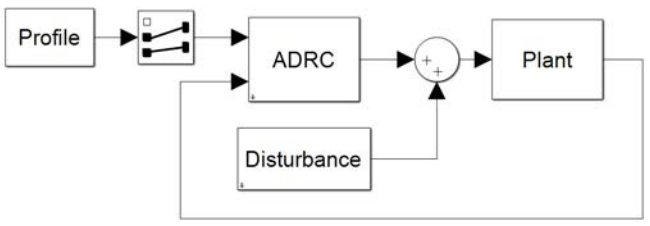

Fig. 3. Simulink Model used in the simulations

The profile block is a 3rd order trajectory generator from [22]. Simulations are performed for both velocity and position control of the plant. Thus, the input profile is either velocity-acceleration, or position-velocity-acceleration. The selector is used for this purpose. The disturbance is a square pulse. Fig.4 shows inside the plant block, modeled in Simscape.

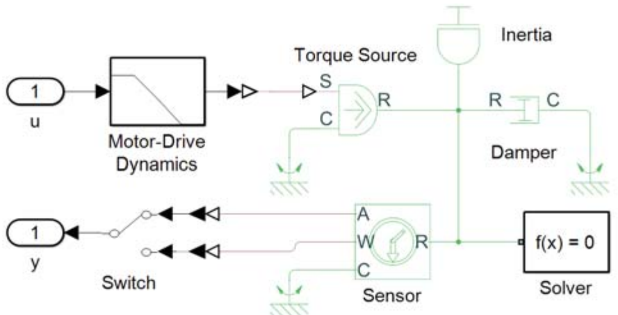

Fig. 4. Plant model used in the simulations

Plant model mainly consists of the following components: First-Order Filter to simulate motor flux and electric drive delays; Ideal Torque Source with "N*m" unit; Inertia; Rotational Damper as viscose friction; Ideal Rotational Motion Sensor with "rad" and "rad/s" feedback sensing units; A Solver Configuration with default parameters. Detailed parameter description for the entire model is described in Table.2.

TABLE II. Model Parameters

Model Part	Block	Parameter	Value	Unit
Profile	3rd Order Profile Generator	Displacement	2	m
		Velocity Bound	1	m/s
		Acceleration Bound	2	m/s^2
		Jerk Bound	3	m/s^3
Controller	ADRC	Input Gain	1/2	-
		Bandwidth	40	rad/s
		System Order	1 or 2	-
		Saturation	3	-
Plant	First-Order Filter	Time Constant	0.01	s
		Filter Type	Lowpass	-
	Inertia	Inertia	2	Kgm^2
	Damper	Damping coefficient	0.2	Nm/(rad/s)
Disturbance	Custom	Duration	0.05	s
		Amplitude	2	Nm
General	All Possible	Sampling Time	0.001	s

In the simulations, system uncertainties include the motor dynamics, which is indeed dominant in many practical systems, and the control signal saturation. Further uncertainties including vibrations due to torque ripples or bad encoder couplings can also be added.

A. *Velocity Control*

First, we set the selector to pass velocity and acceleration. The system order in ADRC is set to 1. A disturbance with the described form is applied at time 2.5 seconds. The speed control curve and the applied torque is shown in Fig.5. The tracking delay due to the control signal saturation is observable clearly.

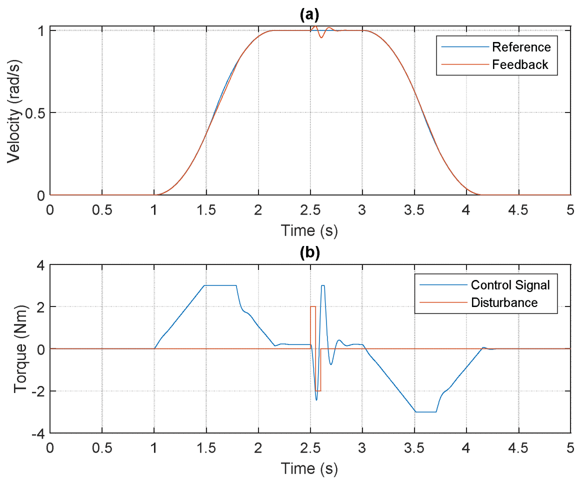

Fig. 5. a) Velocity motion tracking curve showing tracking delays and disturbance (reference vs. actual velocity). b) Torque curve including control signal (which is saturated at 1.5s and 3.5s) and the applied disturbance.

B. *Position Control*

In the second examination, we set the selector to pass position and velocity and acceleration all together. Therefore system order in ADRC should be set to 2. Also, the manual switch in Fig.4 should be switched in such a way to give angle as feedback. The control signal and feedback curves are shown in Fig.6.

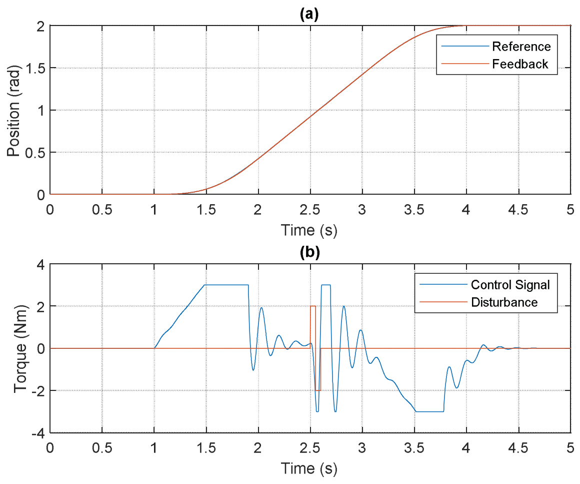

Fig. 6. a) Position motion tracking curve showing reference position vs. actual position. b) Torque curve including control signal and appied disturbance

It is clear that the tracking is perfectly performed, the given disturbance is suitably rejected and the saturation and the system delay uncertainties have minor faulty impacts on the controller performance. The main cause of the oscillations in the control signal is the system delay. The more the delay, the more unstable the whole feedback system gets. System delays will have more effects on controller performance as system order goes higher.

V. Practical Results

In this chapter we intend to show the outcome of all the theories put into practice and see if what claimed, actually happens in practice. The same mechanical plant with more complexities and uncertainties was implemented for the tests, including motor gearbox, initial torque load, coulomb and viscose friction, built-in system resonances and ripples in the feedback signal (due to non-ideal shaft encoder couplings).

A controller based on the presented structure was implemented with 25rad/s bandwidth and 0.25 input gain for the 4Kgm^2 plant inertia. Sampling time in the experiments was 1ms. For most real-time systems a sampling frequency between 0.1-1KHz would suffice. In this experiment no nonlinearity was needed and bandwidth ratio was 1. The plots of the practical results in Fig.7 show the seamless tracking and the controller performance. Compensation of the internal resonance, feedback ripples and massive initial torque load as a disturbance are clearly visible in the torque plot.

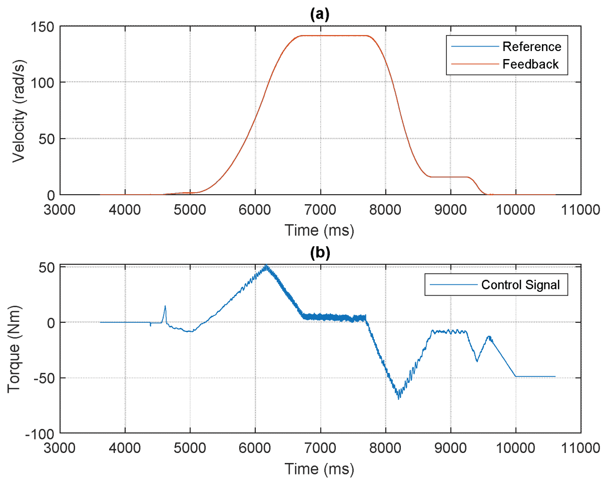

Fig. 7. a) Velocity motion tracking b) Applied control signal

VI. Complementary Materials

A. *Reduced Order Observer*

An important issue in SISO ADRC is how one can use more than one feedback for either ESO or NLSEF. For example, in a position control scenario, how we can put the velocity feedback to a good use as well. In this situation the output state Y is not scalar and the state matrices of (1) will be:

$$A = \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix},\ B = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix},\ C^T = \begin{bmatrix} 1 & 0 \\ 0 & 1 \\ 0 & 0 \end{bmatrix},\ D = 0 \quad (15)$$

Solving $eig(A - LC) = \omega_o$ will result in:

$$L = \begin{bmatrix} 3\omega_o & 0 \\ 3\omega_o^2 & 0 \\ \omega_o^3 & 0 \end{bmatrix} \quad (16)$$

It shows that even if one intends to find the proper observer gains, they would be zero. In such a contradictory case, the designer should reduce the order of ESO by one and give the higher order feedback (position) directly to NLSEF. The predicted disturbance will be according to the measurements of the lower order feedback (velocity). In future designs, reduced order observer concept should be added to ADRC control block.

B. *Bumpless Transfer*

The need to switch between manual control and automatic control in mechatronic systems such as cruise control is obvious. In order to implement Bumpless Transfer in ADRC, all the integrators of ESO must be properly initialized. The current state for every order of the system can be set directly by measuring the outputs. The estimated disturbance (last order of ESO) will be the control signal previously applying to the plant manually multiplied by input gain b. More information on bumpless transfer in ADRC can be found in [23]. Another important note in bumpless transfer is to initialize the input profile so that it can keep the track of the current state of the system. Proper information on how input profiles (trajectory generators) can be appropriately initialized are found in [16-19]. Just like Tracking Mode in PID block controllers, Bumpless Transfer is also a suitable option to be added to ADRC blocks.

VII. Conclusion

This paper describes an optimized structure and principle of ADRC algorithm, and based on Matlab/Simulink, establishes the creation of functional modules of its specific algorithm. Not only it makes the ADRC simulation model simple, but also facilitates parameter tuning through the encapsulation of the modules, thus saving time and greatly improving the modeling and simulation efficiency. The process of subsystem masking, including function modules coding and initializations, is introduced in detail. Moreover, it provides a simple and effective method for simulation and test of the dynamic performance of the system in motion simulations as an example. This model was build based on the analysis of mathematical model of rotational motion platform, using Simscape component of Matlab. Seen from the simulation results, the system can run smoothly and seamlessly, and has good control characteristics. Due to the plant delays, slight fluctuations appeared in the velocity control loop, and as the system order went higher, appeared more substantial fluctuations. The simulations compared with practical results have proven that the modeling method is practical and valid. Finally, important future developments on the controller block have been discussed.

References

[1] Wu, Zhenlong, et al. "Superheated steam temperature control based on modified active disturbance rejection control." *Control Engineering Practice* 83 (2019): 83-97.

[2] Yu, Haibo, et al. "Self-balancing Car Control Based on Active Disturbance Rejection Control (ADRC)." *International conference on Big Data Analytics for Cyber-Physical-Systems*. Springer, Singapore, 2019.

[3] Wang, Gaolin, et al. "Weight-transducerless control strategy based on active disturbance rejection theory for gearless elevator drives." *IET Electric Power Applications* 11.2 (2017): 289-299.

[4] Sun, Li, et al. "Tuning of Active Disturbance Rejection Control with application to power plant furnace regulation." *Control Engineering Practice* 92 (2019): 104122.

[5] Sun, Li, Yuhui Jin, and Fengqi You. "Active disturbance rejection temperature control of open-cathode proton exchange membrane fuel cell." *Applied Energy* 261 (2020): 114381.

[6] Han, Jingqing. "From PID to active disturbance rejection control." *IEEE transactions on Industrial Electronics* 56.3 (2009): 900-906.

[7] Miklosovic, Robert, and Zhiqiang Gao. "A robust two-degree-of-freedom control design technique and its practical application." *Conference Record of the 2004 IEEE Industry Applications Conference, 2004. 39th IAS Annual Meeting.*. Vol. 3. IEEE, 2004.

[8] Chen, Xing, et al. "Tuning method for second-order active disturbance rejection control." *Proceedings of the 30th Chinese Control Conference*. IEEE, 2011.

[9] TEPPA-GARRAN, Pedro Antonio, and Germain Garcia. "Optimal tuning of PI/PID/PID (n-1) controllers in active disturbance rejection control." *Journal of Control Engineering and Applied Informatics* 15.4 (2013): 26-36.

[10] Goforth, Frank J. "On motion control design and tuning techniques." *Proceedings of the 2004 American Control Conference*. Vol. 1. IEEE, 2004.

[11] Gao, Zhiqiang. "Scaling and bandwidth-parameterization based controller tuning." *Proceedings of the American control conference*. Vol. 6. 2006.

[12] Lambrechts, Paul, Matthijs Boerlage, and Maarten Steinbuch. "Trajectory planning and feedforward design for electromechanical motion systems." *Control Engineering Practice* 13.2 (2005): 145-157.

[13] Jiang, Ping, et al. "Modeling and simulation of active-disturbance-rejection controller with simulink." *2010 International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, 2010.

[14] Jiang, Ping, et al. "Modeling and simulation of active-disturbance-rejection controller with simulink." *2010 International Conference on Machine Learning and Cybernetics*. Vol. 2. IEEE, 2010.

[15] HU, HONG-JUN, and ZHOU ZHU. "Modeling and Simulation of Linear Active Disturbance Rejection Controller in Simulink." *DEStech Transactions on Engineering and Technology Research* ecame (2017).

[16] Gao, Zhiqiang. "On discrete time optimal control: A closed-form solution." *Proceedings of the 2004 American Control Conference*. Vol. 1. IEEE, 2004.

[17] Bianco, Corrado Guarino Lo, and Fabio Ghilardelli. "Third order system for the generation of minimum-time trajectories with asymmetric bounds on velocity, acceleration, and jerk." *Matrix* 1.1 (2012): 8.

[18] Bianco, Corrado Guarino Lo, and Friedrich M. Wahl. "A novel second order filter for the real-time trajectory scaling." *2011 IEEE International Conference on Robotics and Automation*. IEEE, 2011.

[19] Bianco, Corrado Guarino Lo, and Fabio Ghilardelli. "A discrete-time filter for the generation of signals with asymmetric and variable bounds on velocity, acceleration, and jerk." *IEEE Transactions on Industrial Electronics* 61.8 (2013): 4115-4125.

[20] Tian, Gang. "Method for automatically estimating inertia in a mechanical system." U.S. Patent No. 8,710,777. 29 Apr. 2014.

[21] Tian, Gang. "Method for automatically estimating inertia, coulomb friction, and viscous friction in a mechanical system." U.S. Patent No. 10,126,202. 13 Nov. 2018.

[22] Paul Lambrechts (2020). Advanced Setpoints for Motion Systems (https://www.mathworks.com/matlabcentral/fileexchange/16352-advanced-setpoints-for-motion-systems), MATLAB Central File Exchange. Retrieved January 31, 2020.

[23] Herbst, Gernot. "Practical active disturbance rejection control: Bumpless transfer, rate limitation, and incremental algorithm." *IEEE Transactions on Industrial Electronics* 63.3 (2015): 1754-1762.