*Article*

# Adaptive and Learning Methods for Drone Motor Control

**Brady Barnett [1], Timothy Sands [2,*]**

[1] Sibley School of Mechanical and Aerospace Engineering, Cornell University, Ithaca, NY 14850 USA
[2] Department of Mechanical Engineering (CVN), Columbia University, New York, NY 10027 USA
* Correspondence: ts2543@caa.columbia.edu

**Abstract:** This manuscript explores unmanned aerial vehicle DC motor control performance efficacy of deterministic artificial intelligence in comparison to model-following adaptation, particularly a direct self-tuner with filtering. The deterministic artificial intelligence model made use of self-awareness statements to overcome error in response to permutations of the multi-duty cycle square wave that served as the system input. It can be seen that (despite equivalently powerful estimation techniques) deterministic artificial intelligence provided far superior results: a reduction in peak initial transient error of 55%, and a mean error reduction over 81% with over 65% reduction in error standard deviation compared to a state-of-the-art nonlinear adaptive control method. Deterministic artificial intelligence also was able to very closely track at the switching of the input control, while the benchmark nonlinear adaptive control failed to respond as quickly at these points.

**Keywords:** drones; artificial intelligence for control; autonomous unmanned aerial vehicles (UAVs); design and modeling; antenna; nonlinear adaptive control; self-tuning regulators

## 1. Introduction

(**a**)

(**b**)

**Figure 1. (a)** Low-cost, low-power, portable mounting platform for auto-tracking antenna, image credit NASA [1] used in accordance with image use policy [2]. (**b**) mounted horn probe antenna, image credit NOAA [3] used in accordance with image use policy [4].

> *Unmanned aerial vehicles, widely known as drones, can act as autonomous communicating nodes, aerial relays, or even aerial base stations, and strongly support the conventional networks in propagation scenarios with obstacles and highly mobile and remote network nodes. [5]*

Lightweight, brushless DC motors optimized for size, weight, and power are often selected for several disparate drone systems. The most common use of motors for drones and unmanned aerial vehicles is to spin the propellers of multirotor drones to enable them to fly. Drone motors may also be found in other unmanned vehicle subsystems, such as camera and payload gimbals, flight surfaces, landing gear, and antenna rotators (the focus of this research). Motor control is a very well researched field and options are proposed in this manuscript directly compared to state-of-the-art methods using numerical figures of merit.

 doi:10.20944/preprints202302.0022.v1



Nonlinear adaptive control is one such well known state-of-the-art field of DC motor control. While the state of nonlinear adaptive control has begun to approach diminishing returns controlling drone DC motors. Meanwhile, other learning systems of control represent an ability to push forth significant performance improvements without the cost of greater computational burden. Following this, the usage of DC motors, and electrical systems in general, continues to grow exponentially as the world moves towards electrification. The ability to create and tune these systems, which are comparatively simple against more advanced nonlinear control methodology, stands to greatly affect the implementation and standard operating procedure of electric motors, and controllable systems in general, in time to come.

The development of adaptive and learning systems has a long history that is presently culminating in a very recent, distinguished lineage in the literature [6, 7, 8, 9], already bestowing three major publication awards, validating the contemporary interest in continued developments. Many techniques are available from this long lineage as alternatives benchmarks for comparing newly proposed methods. Rathmore focused on artificial intelligence applying robust steering control based on tuning of PID controller using the so-called genetic algorithm and the harmonic search algorithm [6]. In 2020, so-called deterministic artificial intelligence was proposed for controlling autonomous unmanned underwater vehicles [7]. The method is based on the self-awareness assertion in the feedforward process dynamics. The feedback signal was formatted by Koo as an adaptive & learning system (proportional derivative feedback and 2-norm optimal least squares). [8] As a duplication and continuation of Koo's work in [8], this manuscript provides an in-depth comparison between the performance of the deterministic artificial intelligence and declared benchmark approaches with iterated step sizes. The performance evaluation mainly focuses on the ability to track the command signals represented by challenging square wave trajectories.

Bernat introduced a speed control approach [10] based on a model-reference adaptive control algorithm for torque load and ripple compensation. Gowri described a direct torque control as one approach focused on discontinuities in rapid modulating commands [11]. Moreover, Rathaiah [12] and Haghi [13] both proposed extremum-seeking adaptive control of first-order systems, which is similar to the approach applied to the vehicle automation control described in [14]. The methods are similarly tested and compared with discontinuous step and square wave inputs. For DC motors, tracking discontinuous step functions and square wave function is challenging since overshoot occurs at the square wave's discontinuities, significantly influencing the tracking performance of nonlinear adaptive control approaches. The difficulty is validated by Vidlak's very contemporary example in Figures 24 and 26 in [15]. This facet is also elaborated by Vidlak's work on tracking performance of self-turning regulators [15] and model reference adaptive control [16].

A similar phenomenon is revealed in Section 3 of this manuscript for the error distribution comparison of different algorithms. This manuscript describes a duplication and continuation of the work presented in [8], which is also work based on its prequel research by Shah [9], where model-following self-turning regulators [17] are chosen as the comparative benchmark approach.

The learning approach evaluated in this manuscript stems from Slotine [18] and the nonlinear adaptive method for robotics [19]. The technique was quickly transformed to expression in coordinates of the body reference frame [20]. In [21], the tunability of feedback and feedforward elements are demonstrated, substantiating the self-awareness statements of deterministic artificial intelligence. [22] As described by Fossen in [23], alternative trajectory tracking control mechanisms based on classical proportional, integral, derivative control; linear-quadratic regulator; feedback linearization; nonlinear backstepping; and sliding mode control are presented. Some approaches are applied to the ocean vehicle mentioned in [24]. The efficacy of such systems is also simulated in [25]. Lorenz and his students also used and evaluated the feedforward element in [26-32]. The fault-

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



tolerance [28], loss reduction [29], loss minimization [30], dead-beat control [31], and self-sensing [32,15] are evaluated and extended from the vehicle to the actuator control circuit. As described in [14,15], the precursor using the physics-based dynamics for virtual sensing, which follows the illustration of optimality and self-sensing, is applied explicitly to DC motors.

*1.1 Literature gaps presented in this manuscript.*

This manuscript presents the following two literature gaps:

1. Development of deterministic artificial intelligence for antenna motor control;
2. Comparative analysis with direct self-tuner with filtering (all process zeros cancelled);

**Main Conclusion of the study**. *Deterministic artificial intelligence was found to outperform a nonlinear adaptive control (direct self-tuning with filtering) with equivalent estimation and computational strength.*

## 2. Materials and Methods

DC motors are well-studied by many methods, providing good comparative benchmarks. Section 2.1 elaborates motor modeling.

*2.1. Motor modeling*

Voltage changes as a function of the current change at the operating point determined by the state of charge, and the voltage open circuit voltage for the equivalent circuit model is determined by a second-order discrete transfer function identical to equation (2) of reference [17]. The model in equation (1) is taken from [17] in discrete form establishing comparative benchmarks in references [9,34].

$$G(z) = \frac{0.09842z + 0.09842}{z^2 - 1.607z + 0.6065} = \frac{Y(z)}{U(z)} \quad (1)$$

**Table 1.** Table of proximal variables and nomenclature [1]

| Variable/acronym | Definition | Variable/acronym | Definition |
|---|---|---|---|
| $G$ | Transfer function | $s$ | Differential variable |
| $Y$ | Output | $z$ | Difference variable |
| $U$ | Input | $t$ | Discrete time variable |

[1] Such tables are offered throughout the manuscript to aid readability.

*2.2. Analytic trajectory generation*

The comparisons presented in this research use a combination of ubiquitous square waves as the input to the system. The respective duty cycles of the waveforms are 50% and 25%, resulting in the net duty cycle being 25%. The waveform was scaled such that the numeric values range from -1 to 1 in amplitude.

*2.3. Deterministic A.I.: self awareness statements using analytic trajectories and PD adaption*

The formulation of the deterministic artificial intelligence model requires restricting the system to respond such that each new timestep corresponds with the self-awareness statement. That is, *u(t)* is calculated according to a *u*(t)* (Equation 2) which binds the input model of the system to a self-tuned system, thus making the system learn. The *u*(t)* values can be defined through the relationship between a matrix of known values [$\Phi_d$], and a vector of unknown values to be solved for $\{\hat{\theta}\}$ initialized with a priori data (Equations 3-5), where equation (5) is derived from the transfer function itself.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



$$u^*(t) \equiv [y_d(t+1) \quad -y_d(t) \quad y_d(t-1) \quad -u_d(t-1)] \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{b}_1 \end{Bmatrix} = [\Phi_d]\{\hat{\theta}\} \tag{2}$$

$$\{\hat{\theta}\} = [\Phi_d]^{-1} u^* \tag{3}$$

$$Error(t) = \{\hat{\theta}\} - y(t) \rightarrow u(t+1) = k_p Error(t) + k_d(\Delta Error(t), Error(t-1)) \tag{4}$$

$$\{\hat{\theta}(0)\} = \begin{Bmatrix} \hat{a}_1 \\ \hat{a}_2 \\ \hat{a}_3 \\ \hat{b}_1 \end{Bmatrix} \approx \begin{Bmatrix} a_1 \\ a_2 \\ a_3 \\ b_1 \end{Bmatrix} \tag{5}$$

It can be seen how the deterministic artificial intelligence utilized the ability to manipulate the model itself according to some estimation (in this case PD or proportional, derivative), allowing the system to self-tune as well as be free from constraints imposed by model-based estimators that are inherently tied to the physical system, instead of being able to adapt towards fitting the input signal best.

**Table 2.** Table of proximal variables and nomenclature [1]

| Variable/acronym | Definition | Variable/acronym | Definition |
|---|---|---|---|
| $u^*$ | Control input | $\Phi_d$ | Regressor matrix |
| $y_d$ | Desired output | $\hat{\theta}$ | Parameter vector |
| $\hat{a}_1, \hat{a}_2, \hat{a}_3, \hat{b}_1$ | Estimates | $a_1, a_2, a_3, b_1$ | True values |
| $k_p$ | Proportional gain | $k_d$ | Difference gain |
| $A$ | Coefficients of $U(z)$ | $B$ | Coefficients of $Y(z)$ |

[1] Such tables are offered throughout the manuscript to aid readability.

## 3. Results

Section 2 outlined the establishment of the model and estimators working on the deterministic artificial intelligence methodology, which will now be compared against a direct self-tuner with filtering (with all process zeros cancelled).

### *3.1. Direct self-tuner with filtering results*

Section 2 outlined the establishment of the model and estimators working on the deterministic artificial intelligence methodology, which will now be compared against a direct self-tuner with filtering (with all process zeros cancelled). The state-of-the-art benchmark method is well known [9,17,33,34] and is briefly summarized in this section.

The direct self-tuner aims to reparametrize the model in terms of the controller parameters, demonstrated in the convergence of the new parameters shown in Figure 1b. The process of doing such is outlined in Section 3.5 of [2], and is done through use of the Diophantine equation to break down the reparameterization into the following pseudo-code:

1. Calculate terms in denominator $A = [\hat{a}_1 \quad \hat{a}_2 \quad \hat{a}_3]$ with use of estimator.
2. Declare "left-over" A terms.
3. Establish left-over terms from factoring numerator $B^+$ become $R'B^+$.
4. Combine $B^-$ gain terms with that cannot be cancelled in $B = [\hat{b}_1]$.
5. Output the polynomial of the controller.

The nonlinear adaptive controller, due to the coefficients of the polynomial being parameterized and solved for, provides estimation without manual tuning.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



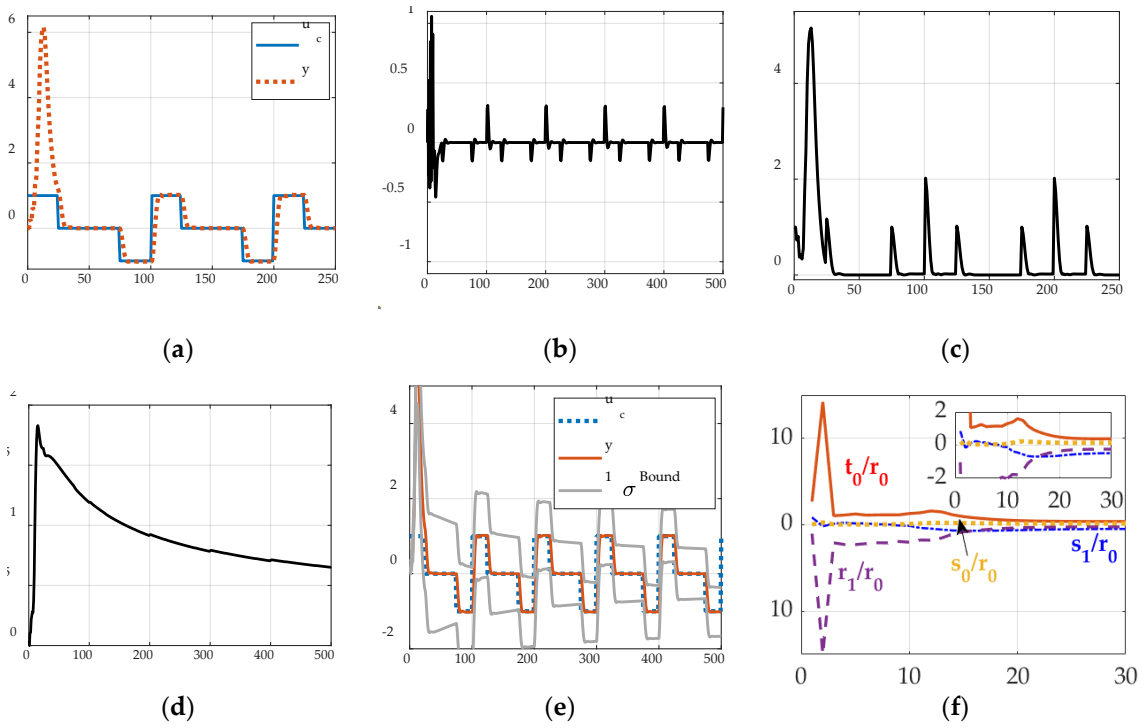



**Figure 1.** Results plotted versus time in seconds on the abscissa. Direct self-tuner with filtering with process zero cancellation. **(a)** response to input, (**b**) $u$ response, (**c**) error, (**d**) standard deviation; (**e**) response to input, (**f**) normalized parameter estimates

Figure 1a shows the large overshoot at the initial displacement condition, and while the system does mitigate error over time, even after the full propagation of the system, significant error still exists. It can be seen that at all switches, the system responds fairly slowly, but with minimal overshoot. The convergence of the parameter estimates does occur rapidly, reaching convergence at around 20 timesteps, or at around 4% of the runtime of this simulation.

**Table 3.** Figures of merit corresponding to figure 1 direct self-tuner with process zero cancellation [1]

| Data source | Mean | Standard deviation |
|---|---|---|
| Subfigure (**a**) command | 0.0059 | 0.7113 |
| Subfigure (**b**) output | -2e-04 | 0.1047 |
| Subfigure (**c**) command | 0.2272 | 0.6555 |
| Subfigure (**d**) command | 0.9187 | 0.3064 |
| Subfigure (**e**) input command | 0.0059 | 0.7113 |
| Subfigure (**e**) output | 0.0893 | 0.9875 |
| Subfigure (**e**) 1-$\sigma$ bounds | 1.0080 | 1.1024 |
| Subfigure (**f**) $t_0/r_0$ | 0.1434 | 0.0158 |
| Subfigure (**f**) $r_1/r_0$ | -0.417 | 0.0991 |
| Subfigure (**f**) $s_0/r_0$ | 0.4280 | 0.6408 |
| Subfigure (**f**) $s_1/r_0$ | -0.305 | 0.7094 |

[1] Such tables are offered throughout the manuscript to aid readability.

### *3.2. Deterministic artificial intelligence*

The responses generated were according to the model and estimations described in Section 2.3.

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



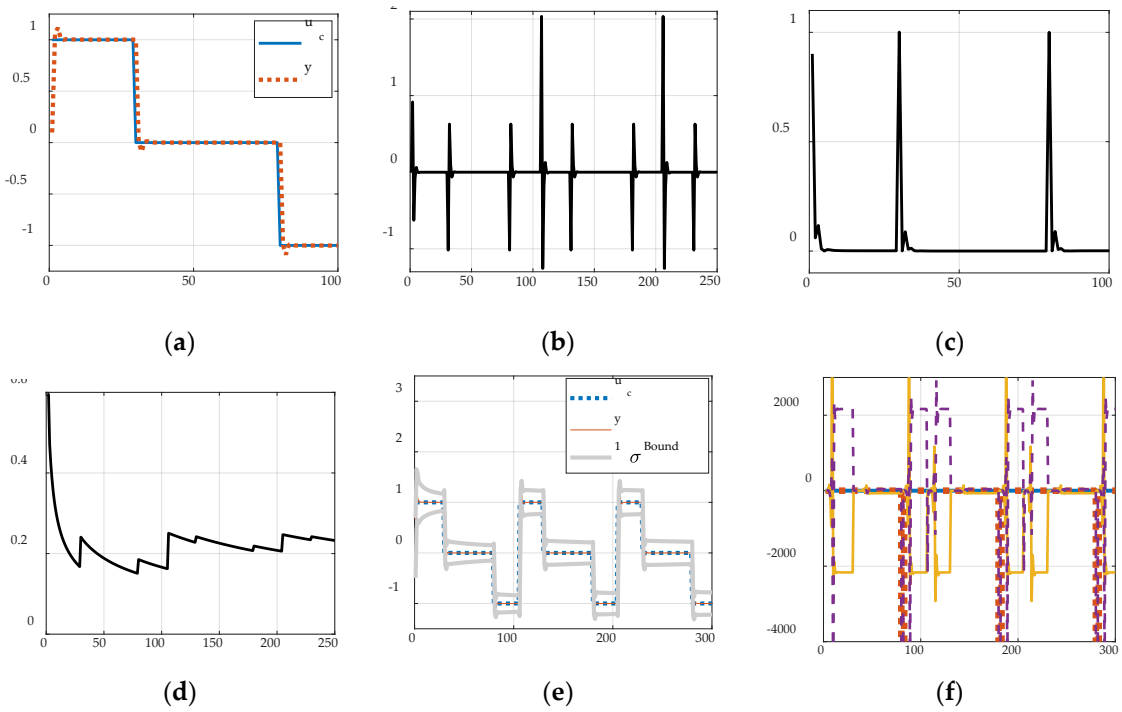
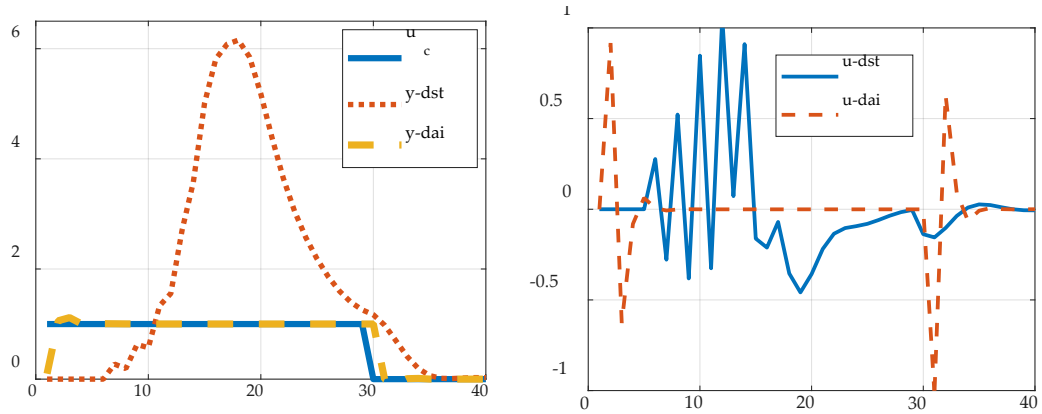



**Figure 2.** Results plotted versus time in seconds on the abscissa. Deterministic artificial intelligence results. (a) Command and response (top, left), input (top, right), error over time (bottom, left), and standard deviation over time (bottom, right). (b) Response to command with standard deviation bands (top) and normalized, converging parameter estimates over time, note logarithmic y-axis due to the magnitude of values (bottom) (a) Response to input, (b) U response, (c) Error, (d) Standard deviation; (e) Response to input, (f) Normalized parameter estimates

It can be seen that the deterministic artificial intelligence system has significantly less overshoot, as well as almost negligible error in regions immediately away from the switches. While the initial standard deviation is already far lower than the previous methodology in Section 3.1, the decay of the standard deviation is also greater. It should be noted that the proportional and derivative gain of the system were lightly tuned through simple looping while looking to minimize the total error of the system, resulting in $K_p = 0.3887, K_d = 0.6279$.

### *3.3. Comparison of Estimation Accuracy*

This section compares the numerical error performance between the two methods. Figure 3 shows the comparison in the raw results, where the difference around switches can be highlighted as the greatest strength of the deterministic artificial intelligence method.



Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



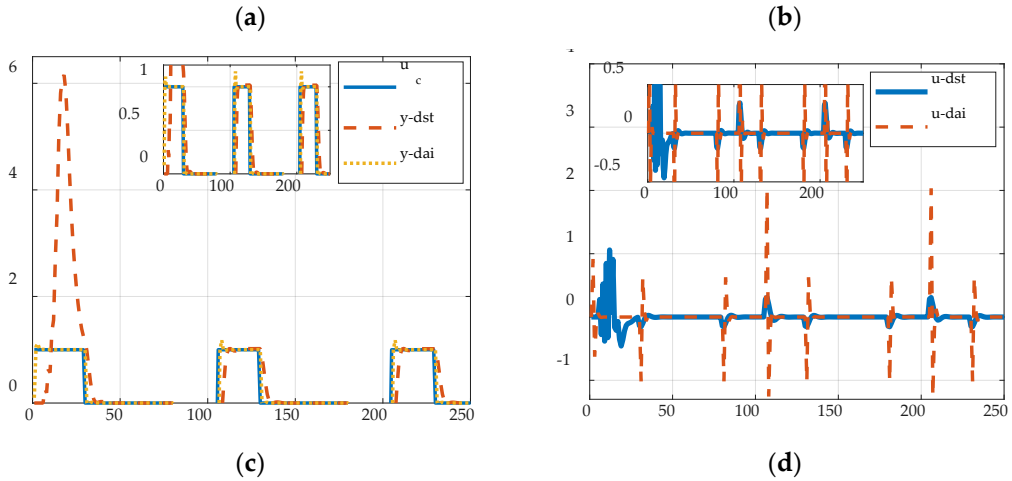



**Figure 3.** Results plotted versus time in seconds on the abscissa. Results. (a) Command and response in initial region (top, left), input in initial region (top, right) command and response over time (bottom, left), and input over time (bottom, right)

**Table 3.** Criteria Comparison Between Methods. [1]

| Method | Initial error | Tracking error | |
|---|---|---|---|
| | | Mean, $\mu$ | Standard deviation, $\sigma$ |
| Direct self-tuner with filtering (all process zeros cancelled) | 2.0209069 | 0.22724 | 0.65545 |
| Deterministic artificial intelligence | **0.90158** | **0.042619** | **0.2290** |

[1] Such tables are offered throughout the manuscript to aid readability.

The results in bold font within Table 3 indicate that deterministic artificial intelligence performed better across all three metrics analyzed, as well as Figure 3 shows visually the superior performance in terms of transient behavior, especially in terms of most accurately capturing the command signal.

## 4. Discussion

It can be seen that the deterministic artificial intelligence had superior performance across almost every metric and given that the actual estimation within the deterministic artificial intelligence was done with equivalently powerful estimators, it can be shown clearly that the ability to affect the model greatly contributes to improved performance. The greatest points of interest come in response to the initial condition and at switches. Within the initial region, almost no overshoot is experienced, which is where the 55% reduction in peak accuracy comes, as the non-learning controller overshoots so drastically.

**Table 4.** Percent performance improvement comparisons. [1]

| Method | Initial error | Tracking error | |
|---|---|---|---|
| | | Mean, $\mu$ | Standard deviation, $\sigma$ |
| Deterministic artificial intelligence | –55% | –81% | –65% |

[1] Such tables are offered throughout the manuscript to aid readability.

The ability to pivot immediately in response to a single timestep at the switches to capture the command behavior also stands out, as no traditional controller can respond to this quickly, due to the system inertia present in any plant. By overriding this, the controller is able to radically pivot. It should be noted, however, that given the massive spike in the adjusted parameters, this behavior may be unrealistic due to the physical limitations of a system: the massive spike could lead to an unachievable gain and simply require too

 doi:10.20944/preprints202302.0022.v1



much power to ever happen. Further research into fine-tuning the model for a DC motor could reveal more about these constraints, as well as other concerns even as simple as the amount of stress induced by such an abrupt response.

**Author Contributions:** Conceptualization, B.B. and T.S.; methodology, B.B. and T.S.; software, B.B. and T.S.; validation, B.B. and T.S.; formal analysis, B.B.; investigation, B.B. and T.S.; resources, T.S.; data curation, B.B.; writing—original draft preparation, B.B.; writing—review and editing, B.B. and T.S.; visualization, B.B. and T.S.; supervision, T.S.; project administration, T.S.; funding acquisition, T.S. All authors have read and agreed to the published version of the manuscript. Please turn to the CRediT taxonomy for the term explanation. Authorship must be limited to those who have contributed substantially to the work reported.

**Funding:** Please add: This research received no external funding. The APC was funded by the corresponding author.

**Data Availability Statement:** Data supporting reported results can be provided by contacting the author.

**Conflicts of Interest:** The authors declare no conflict of interest.

## Appendix A

MATLAB Code used to generate results.

1. Direct Self-Tuning with Filtering Script

```
% Deterministic AI Script
clear all; close all; clc;
%Plant Characteristics
B = [0 0.9842 0.09842]; A = [1 -1.607 0.6065];
Bm=[0 0.1065 0.092]; Am=[1 -1.3205 0.4966];
a0=1.0;d0=1;am0=Am(1);am1=Am(2);am2=Am(3); n=4; lambda=1;

%Initialize
nzeros=5;time=zeros(1,nzeros);maxtime=500;
Y=zeros(1,maxtime+nzeros);U=zeros(1,maxtime+nzeros); Uf=ones(1,nzeros);Yf=ones(1,nzeros);

r0=5; r1=0.85*r0; s0=2.68*r0; s1=-1.03*r0; t0=1.65*r0; P=100*eye(n); THETA_hat(:,1)=[r0 r1 s0 s1]';

%Square wave with 50 sec period
for i=1:maxtime
    Uc(i)=(square(2*pi/50*i)+square(pi/50*i))/2;
end

Uc=[ones(1,nzeros),Uc]; Uc(1,100:105)=-1;

for i=1:maxtime
   phi=[];t=i+nzeros;time(t)=i;

   %Y(t)=[r0 r1 s0 s1]*[Uf(t-d0) Uf(t-d0-1) Yf(t-d0) Yf(t-d0-1)]';
   Y(t)=[-A(2) -A(3) B(2) B(3)]*[Y(t-d0) Y(t-d0-1) U(t-d0) U(t-d0-1)]';

   % RLS-EF algorithm on model 3.30 to estimate R & S coefficients, calculate T
   phi=[Uf(t-d0) Uf(t-d0-1) Yf(t-d0) Yf(t-d0-1)]'; K=P*phi/(lambda+phi'*P*phi);
   P=P-P*phi/(1+phi'*P*phi)*phi'*P/lambda; error(i)=Y(t)-phi'*THETA_hat(:,i);
   THETA_hat(:,i+1)=THETA_hat(:,i)+K*error(i);
   r0=THETA_hat(1,i+1); r1=THETA_hat(2,i+1); s0=THETA_hat(3,i+1); s1=THETA_hat(4,i+1);
   t0=[1+am1+am2];

   % Calculate Model control
   U(t)=[-r1 t0 -s0 -s1]*[U(t-d0) Uc(t) Y(t) Y(t-d0)]'/r0;

   % Calculate filtered output and control
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



```
    Yf(t)=[1 -am1 -am2]*[Y(t) Yf(t-d0) Yf(t-d0-1)]';   Uf(t)=[1 -am1 -am2]*[U(t) Uf(t-d0) Uf(t-d0-1)]';
end

%Plot
error = zeros(1,length(Y));
for i=1:1:length(Y);   error(i) = abs(Y(i)-Uc(i)); end
sd_set = zeros(1,length(Y));
for i=2:1:length(error);   sd_set(i) = std(error(1:i)); end
sd_set(1) = sd_set(2);

[peak_transient_error, pos] = max(abs(Y(50:end)-Uc(50:end)));
%peak_transient_error_percent = abs(Y(pos))/abs(Uc(pos));

[peak_input_error, pos] = max(abs(Y(1:20)-Uc(1:20)));
peak_input_error_percent = abs(Y(pos))/abs(Uc(pos));
error_mean = mean(error(:));

%Response Plot
figure(1);
subplot(2,2,1);plot(time,Uc,time,Y);axis([0 250,-1.25 4]);
xlabel('Timesteps','fontsize',12);legend('u_c','y');
title('Response to Input','fontsize',13);
subplot(2,2,2);plot(time,U);xlabel('Timesteps','fontsize',12);axis([0 500 -2.7 2.7]);
title('U Response','fontsize',13);
subplot(2,2,3);plot(time,error);xlabel('Timesteps','fontsize',12);axis([0 250,-0.1 4]);
title('Error','fontsize',13);
subplot(2,2,4);plot(time,sd_set);xlabel('Timesteps','fontsize',12);axis([0 500 0 2]);
title('Standard Deviation','fontsize',13);

%Fancy Response Mode
sd_up = Y+sd_set; sd_low = Y-sd_set;
figure(2);
subplot(2,1,1);plot(time,Uc,'--',time,Y,time,sd_up,'r',time,sd_low,'r');
axis([0 500,-2 3]);xlabel('Timesteps','fontsize',12);legend('u_c','y','1 \sigma Bound');
title('Response to Input','fontsize',13);

R0=THETA_hat(1,:);R1=THETA_hat(2,:);S0=THETA_hat(3,:);S1=THETA_hat(4,:);
for i=1:501;t(i)=i;end
subplot(2,1,2);plot(t,R1./R0,t,S0./R0,t,t0./R0,t,S1./R0);xlabel('Timesteps','fontsize',14);
axis([0 30 -4 4]); title(' Normalized Parameter Estimates','fontsize',13);
text(11,1.75,'t_0/r_0','fontsize',14);text(12.5,0.25,'s_0/r_0','fontsize',13);
text(11.5,-1.25,'r_1/r_0','fontsize',14);text(15,-1.25,'s_1/r_0','fontsize',13);

%Plant Functions
maxtime = 500;
B = [0 0.9842 0.09842]; A = [1 -1.607 0.6065]; Bm=[0 0.1065 0.092]; Am=[1 -1.3205 0.4966];

H=tf(Bm,Am,0.1); %Convert Plant [num] and [den] to discrete transfer function
Rmatrix=[];

n=4;lambda=1; % number of parameters to estimate and Exponential Forgetting Factor
nzeros=5; time=zeros(1,nzeros); Y=zeros(1,nzeros); Ym=zeros(1,nzeros);%Initialize ouput vectors
U=ones(1,nzeros); error_collect=ones(1,nzeros);
Noise = 1/25*randn(1,maxtime+nzeros); epsilon=[zeros(1,nzeros+maxtime)];

%Square wave with 50 sec period
for l=1:maxtime; Uc(l)=(square(2*pi/50*l)+square(pi/50*l))/2; end

Uc=[ones(1,nzeros),Uc]; Uc(1,100:105)=-1; phi_awr = [];
```

Preprints (www.preprints.org) | NOT PEER-REVIEWED | Posted: 2 February 2023 doi:10.20944/preprints202302.0022.v1



```
t=1:1:maxtime;
traj_Uc = (Uc); hvy_m = [zeros(1,nzeros) traj_Uc]; eb = Y(1) - hvy_m(1);
err = 0; kp = 0.3887; kd = 0.6279; phid = []; ustar = []; hatvec = zeros(4,1);

for i=1:maxtime
    t=i+nzeros; time(t)=i; de = err-eb; u = kp*err + kd*de; U(t-1) = u;

    Y(t)=[-A(2) -A(3) B(2) B(3)]*[Y(t-1) Y(t-2) U(t-1) U(t-2)]';

    phid = [phid; Y(t) -Y(t-1) Y(t-2) -U(t-2)]; ustar = [ustar; u];
    phid_t = phid(i,:); ustar_t = u;
    newest = phid_t.\ustar_t; hatvec(:,i) = newest; eb = err; err = hvy_m(t)-Y(t);
end

time = time((nzeros+1):end); Y = Y((nzeros+1):end); U = U((nzeros):end); Uc = Uc(1:length(time));
error = zeros(1,length(Y));
for i=1:1:length(Y); error(i) = abs(Y(i)-Uc(i)); end
sd_set = zeros(1,length(Y));
for i=2:1:length(error); sd_set(i) = std(error(1:i)); end
sd_set(1) = sd_set(2);

THETA_hat = [hatvec(2,:)./hatvec(1,:); hatvec(3,:)./hatvec(1,:);
ones(1,length(hatvec(1,:)))./hatvec(1,:); hatvec(4,:)./hatvec(1,:)];

%Response Plot
figure(1);
subplot(2,2,1);plot(time,Uc,time,Y);axis([0 100,-1.25 1.25]);
xlabel('Timesteps','fontsize',12);legend('u_c','y');
title('Response to Input','fontsize',13);
subplot(2,2,2);plot(time,U);xlabel('Timesteps','fontsize',12);axis([0 250 -2.7 2.7]);
title('U Response','fontsize',13);
subplot(2,2,3);plot(time,error);xlabel('Timesteps','fontsize',12);axis([0 100,-0.1 1.25]);
title('Error','fontsize',13);
subplot(2,2,4);plot(time,sd_set);xlabel('Timesteps','fontsize',12);axis([0 250 0 0.75]);
title('Standard Deviation','fontsize',13);

[peak_transient_error, pos] = max(abs(Y(50:end)-Uc(50:end)));
peak_transient_error_percent = abs(Y(pos))/abs(Uc(pos));

[peak_input_error, pos] = max(abs(Y(1:24)-Uc(1:24)));
peak_input_error_percent = abs(Y(pos))/abs(Uc(pos));
error_mean = mean(error(:));

%Fancy Response Mode
sd_up = Y+sd_set; sd_low = Y-sd_set;
figure(2);
subplot(2,1,1);plot(time,Uc,'--',time,Y,time,sd_up,'r',time,sd_low,'r');axis([0 300,-1.75 1.75]);
xlabel('Timesteps','fontsize',12);legend('u_c','y','1 \sigma Bound');
title('Response to Input','fontsize',13);

for i=1:1:4; THETA_hat(1,:) = sign(THETA_hat(1,:)).*log10(abs(THETA_hat(1,:))); end

R0=THETA_hat(1,:);R1=THETA_hat(2,:);S0=THETA_hat(3,:);S1=THETA_hat(4,:);
for i=1:501; t(i)=i; end

subplot(2,1,2);plot( time,R0,time,R1,time,S0,time,S1);
xlabel('Timesteps','fontsize',14);axis([0 300 -3000 3000]); ylabel('Logarithmic Scale')
title('Automated System Parameters','fontsize',13); legend('a1','a2','a3','b1');
```

## References

1. Low-Cost, Low-Power, Portable Mounting Platform for Auto-Tracking Antenna. Available online: https://www.nasa.gov/sites/default/files/files/Low-Cost_Low-Power_Portable_Mounting_Platform_for_Auto-Tracking_Antenna-20150703-LOW.pdf (accessed on 15 January 2023)
2. NASA Image Use Policy. Available online: https://gpm.nasa.gov/image-use-policy (accessed on 24 December 2022).
3. Umeyama, A.; Salazar-Cerreno, J.; Fulton, C. UAV-Based Antenna Measurements for Polarimetric Weather Radars: Probe Analysis. *IEEE Access*, **2020**, *8*, 191862-191874.
4. NOAA Image Use Policy. Available online: https://www.fisheries.noaa.gov/national/about-us/website-policies-and-disclaimers#:~:text=Any%20NOAA%20images%20on%20our,%E2%80%9CNOAA%E2%80%9D%20as%20the%20source (accessed on 24 December 2022).
5. Special Issue "Advances of Unmanned Aerial Vehicle Communication". Available online: https://www.mdpi.com/journal/drones/special_issues/KEM553JC45 (accessed on 15 January 2023).
6. Rathore, Ankush and Mahendra Kumar. Robust Steering Control of Autonomous Underwater Vehicle: based on PID Tuning Evolutionary Optimization Technique. *International Journal of Computer Applications* **2015**, 117, 1-6.
7. Sands, T. Development of deterministic artificial intelligence for unmanned underwater vehicles (UUV). *J. Mar. Sci. Eng.* **2020**, 8, 578.
8. Koo, S.M.; Travis, H.; Sands, T. Impacts of Discretization and Numerical Propagation on the Ability to Follow Challenging Square Wave Commands. *J. Mar. Sci. Eng.* **2022**, 10, 419. https://doi.org/10.3390/jmse10030419
9. Shah, R.; Sands, T. Comparing Methods of DC Motor Control for UUVs. *Appl. Sci.* **2021**, 11, 4972.
10. Bernat, J.; Stepien, S. The adaptive speed controller for the BLDC motor using MRAC technique. *IFAC Proc.* Vol. **2011**, 44, 4143–4148.
11. Gowri, K.; Reddy, T.; Babu, C. Direct torque control of induction motor based on advanced discontinuous PWM algorithm for reduced current ripple. *Electr. Eng.* **2010**, 92, 245–255.
12. Rathaiah, M.; Reddy, R.; Anjaneyulu, K. Design of Optimum Adaptive Control for DC Motor. *Int. J. Electr. Eng.* **2014**, 7, 353–366.
13. Haghi, P.; Ariyur, K. Adaptive First Order Nonlinear Systems Using Extremum Seeking. In Proceedings of the 50th Annual Allerton Conference on Communication Control, Monticello, IL, USA, 1–5 October **2012**; pp. 1510–1516.
14. Sands, T. Virtual sensoring of motion using Pontryagin's treatment of Hamiltonian systems. Sensors **2021**, 21, 4603.
15. Vidlak, M.; Gorel, L.; Makys, P.; Stano, M. Sensorless Speed Control of Brushed DC Motor Based at New Current Ripple Component Signal Processing. *Energies* **2021**, *14*, 5359.
16. Chen, J.; Wang, J.; Wang, W. Robust Adaptive Control for Nonlinear Aircraft System with Uncertainties. *Appl. Sci.* **2020**, 10, 4270.
17. Åström, K.;Wittenmark, B. Adaptive Control; Addison-Wesley: Boston, FL, USA, **1995**.
18. Slotine, J.; Benedetto, M. Hamiltonian adaptive control on spacecraft. *IEEE Trans. Autom.* Control **1990**, 35, 848–852.
19. Slotine, J.;Weiping, L. Applied Nonlinear Control; Prentice Hall: Englewood Cliffs, NJ, USA, **1991**.
20. Fossen, T. Comments on "Hamiltonian Adaptive Control of Spacecraft". *IEEE Trans. Autom.* Control **1993**, 38, 671–672.
21. Sands, T.; Kim, J.J.; Agrawal, B.N. Improved Hamiltonian adaptive control of spacecraft. In Proceedings of the IEEE Aerospace, Big Sky, MT, USA, 7–14 March 2009; IEEE Publishing: Piscataway, NJ, USA, **2009**; pp. 1–10.
22. Smeresky, B.; Rizzo, A.; Sands, T. Optimal Learning and Self-Awareness Versus PDI. Algorithms **2020**, 13, 23.
23. Fossen, T. Handbook of Marine Craft Hydrodynamics and Motion Control, 2nd ed.; John Wiley & Sons Inc.: Hoboken, NJ, USA, 2021; ISBN 978-1-119-57505-4.
24. Fossen, T. Guidance and Control of Ocean Vehicles; John Wiley & Sons Inc.: Chichester, UK, **1994**.
25. Sands, T.; Bollino, K.; Kaminer, I.; Healey, A. Autonomous Minimum Safe Distance Maintenance from Submersed Obstacles in Ocean Currents. *J. Mar. Sci. Eng.* **2018**, 6, 98.
26. Sands, T.; Lorenz, R. Physics-Based Automated Control of Spacecraft. In Proceedings of the AIAA Space Conference & Exposition, Pasadena, CA, USA, 14–17 September 2009.
27. Available online: https://site.ieee.org/ias-idc/2019/01/29/prof-bob-lorenz-passed-away/ (accessed on 12 Dec 2022).
28. Zhang, L.; Fan, Y.; Cui, R.; Lorenz, R.; Cheng, M. Fault-Tolerant Direct Torque Control of Five-Phase FTFSCW-IPM Motor Based on Analogous Three-phase SVPWM for Electric Vehicle Applications. *IEEE Trans. Veh. Technol.* **2018**, 67, 910–919.
29. Apoorva, A.; Erato, D.; Lorenz, R. Enabling Driving Cycle Loss Reduction in Variable Flux PMSMs Via Closed-LoopMagnetization State Control. *IEEE Trans. Ind. Appl.* **2018**, 54, 3350–3359.
30. Flieh, H.; Lorenz, R.; Totoki, E.; Yamaguchi, S.; Nakamura, Y. Investigation of Different Servo Motor Designs for Servo Cycle Operations and Loss Minimizing Control Performance. *IEEE Trans. Ind. Appl.* **2018**, 54, 5791–5801.
31. Flieh, H.; Lorenz, R.; Totoki, E.; Yamaguchi, S.; Nakamura, Y. Dynamic Loss Minimizing Control of a Permanent Magnet Servomotor Operating Even at the Voltage Limit When Using Deadbeat-Direct Torque and Flux Control. *IEEE Trans. Ind. Appl.* **2019**, 3, 2710–2720.
32. Flieh, H.; Slininger, T.; Lorenz, R.; Totoki, E. Self-Sensing via Flux Injection with Rapid Servo Dynamics Including a Smooth Transition to Back-EMF Tracking Self-Sensing. *IEEE Trans. Ind. Appl.* **2020**, 56, 2673–2684.
33. Sands, T. Comparison and Interpretation Methods for Predictive Control of Mechanics. Algorithms **2019**, 12, 232.
34. Sands, T. Control of DC Motors to Guide Unmanned Underwater Vehicles. *Appl. Sci.* **2021**, *11*, 2144.
35.

36.
37. Author 1, A.B.; Author 2, C.D. Title of the article. *Abbreviated Journal Name* **Year**, *Volume*, page range.
38. Author 1, A.; Author 2, B. Title of the chapter. In *Book Title*, 2nd ed.; Editor 1, A., Editor 2, B., Eds.; Publisher: Publisher Location, Country, 2007; Volume 3, pp. 154–196.
39. Author 1, A.; Author 2, B. *Book Title*, 3rd ed.; Publisher: Publisher Location, Country, 2008; pp. 154–196.
40. Author 1, A.B.; Author 2, C. Title of Unpublished Work. *Abbreviated Journal Name* year, *phrase indicating stage of publication (submitted; accepted; in press)*.
41. Author 1, A.B. (University, City, State, Country); Author 2, C. (Institute, City, State, Country). Personal communication, 2012.
42. Author 1, A.B.; Author 2, C.D.; Author 3, E.F. Title of Presentation. In Proceedings of the Name of the Conference, Location of Conference, Country, Date of Conference (Day Month Year).
43. Author 1, A.B. Title of Thesis. Level of Thesis, Degree-Granting University, Location of University, Date of Completion.
44. Title of Site. Available online: URL (accessed on Day Month Year).