Introduction

This exercise was devoted to the adaptive control design problem in the **MIAC** (Model-Identification Adaptive Control) scheme for the **aero-dynamic plant (specific to its rolling dynamics)** and to verification of the designed control system in the **Matlab-Simulink environment**. We applied a **deterministic** approach to the controller synthesis, which was reasonable under assumption that the stochastic noise disturbing the plant is negligibly small (**signal-to-noise ratio** was sufficiently high).

Description of the Plant:

We considered the rolling dynamics of the aero-plant which can be controlled by adjusting the roll angle.

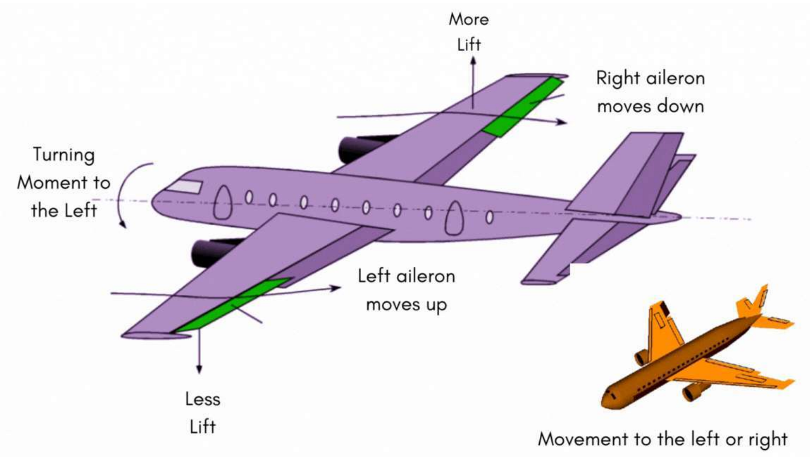


Fig1: Aileron roll dynamics | Sources: [Aircraft Maneuverability and Stability - Global Aviation]

It is possible to change the aircraft roll-rate ω expressed in [rad/s]. Locally (in a small vicinity of zero steady-state conditions), we can approximate the roll-rate dynamics by the following linear differential equation: $\mathbf{\dot{\omega} = -a_0\omega + b_0\delta a,}$

we can rewrite the above equation as:

$$T_0\dot{y} + y = k_0u, ------ where : y = \omega, u = \delta_a, T_0 = 1/a_0, k_0 = b_0/a_0$$

p = [T, k]'

where we assumed that only input **u** and output **y** are available for measurements, while T0 and k0 denote, respectively, the **true time-constant and the true dc-gain of the plant** (unknown in practice and possibly time-varying). The values of control input u = δa were inherently constrained to the range [−π; π] rad due to physical interpretation of δa.

Since the true parameter values were not known, we used the parameter identification scheme to predict the real parameter values.

This predication was coupled with PI controller through parameter synthesising, that formed the Model Identification Adaptive Control (MIAC) scheme.

Objective of the Task

The objective of this control method was to predict the parameter values online that guarantees:

R1. The time varying reference trajectory and its derivative to be exist and bounded

R2. Tracking error asymptotically approaches to zero as time goes infinity with no overshot

R3. Settling time must be in 1% vicinity range

Control Method and Description

MIAC scheme:

MIAC is a type of adaptive control that provides a framework for the automatic, real-time adjustment of controllers based on the estimation of the current state of the controlled plant. It is also called self-Tuning Adaptive Control.

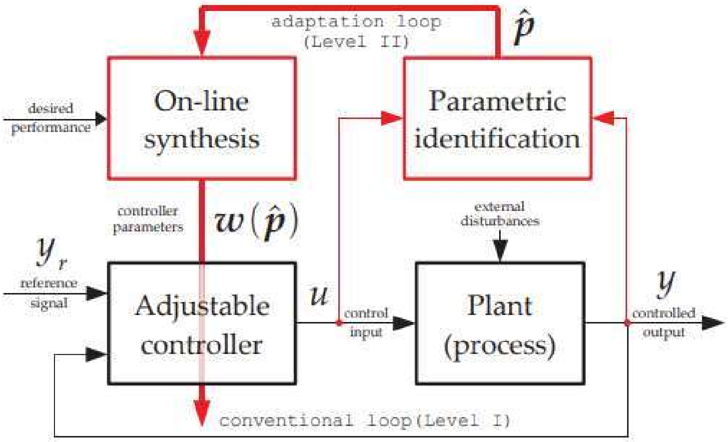


Fig2: Working principle of MIAC => Sources: lecture material

- $\omega(\hat{p})$ - is the output of the parameter synthesis block that used to change the controller parameter online
- $\hat{p}$ - is the parameter computed by the parameter identification block.
- Conventional loop - the feedback loop with sampling time Tc
- Adaptation loop - the new block that compu tes the parameters. Its sampling time Ta is larger than Tc.

Control System Design of MIAC

Design stage

1. Assumed the model structure **ARX** with particular degrees of polynomials; then selected and prepared the recursive estimator of parameters **SVF-RLS.**

2. Prescribe the desired control performance for a closed-loop system $\Rightarrow$ select a controller structure and quantitative criteria,

$$e(\infty) = 0, \quad T_{s1\%} < 1\,\text{s}, \qquad \Rightarrow \quad \text{PI cont. with pole-placement}$$

3. Derive analytical synthesis equations for controller parameters $\boldsymbol{w}$ using criteria from point 2 (express parameters $\boldsymbol{w}$ as a function of plant parameters $\boldsymbol{p}_o$),

$$\text{for PI controller}: \ \boldsymbol{w}(\boldsymbol{p}_o) = [k_p(\boldsymbol{p}_o) \quad T_i(\boldsymbol{p}_o)]^\top$$

4. In the derived synthesis equations replace the *true* plant parameters $\boldsymbol{p}_o$ with their estimates $\hat{\boldsymbol{p}}$ according to the CE principle (because $\boldsymbol{p}_o$ is unknown in practice):

$$\text{for PI controller}: \ \boldsymbol{w}(\hat{\boldsymbol{p}}) = [k_p(\hat{\boldsymbol{p}}) \quad T_i(\hat{\boldsymbol{p}})]^\top$$

The results from points 1 to 4 can be utilized on-line in the MIAC scheme

On-line computation stage

S1. Compute current estimates of plant parameters:

$$\hat{\boldsymbol{p}}(nT_a) = [\hat{p}_1(nT_a)\ \hat{p}_2(nT_a)\ \ldots\ \hat{p}_d(nT_a)]^\top \tag{1}$$

S2. Update controller parameters:

$$\boldsymbol{w}(nT_a) := \boldsymbol{w}(\hat{\boldsymbol{p}}(nT_a)) \tag{2}$$

S3. Compute a current control input using the updated controller parameters:

$$u(nT_c) := g(\boldsymbol{w}(nT_a), y_r(nT_c), y(nT_c), \ldots) \tag{3}$$

S4. GOTO **S1.** or **S3.**

Simulation Results and Analysis

Simulink Model

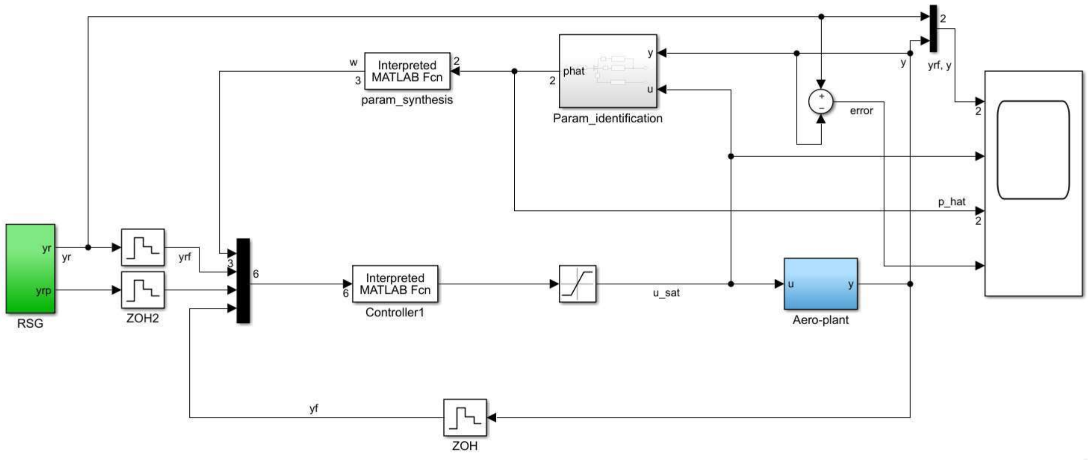


Fig3: Complete simulink model

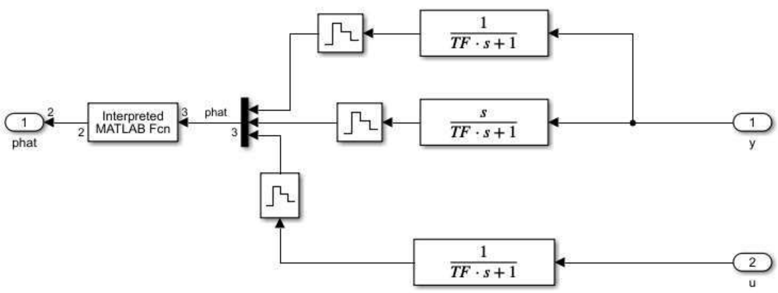


Fig4: Parameter synthesis block with first order SVF

Results and Analysis

In this section, we discussed the open loop and closed loop analysis separately.

i. Open loop analysis

- Reference signal: sinusoidal with amplitude **1rad/s** and wr = **0.5rad/s**
- simulation run time: **50sec**
- Ta = 50Tc
- P = 1000*I ==> I = eye(2)

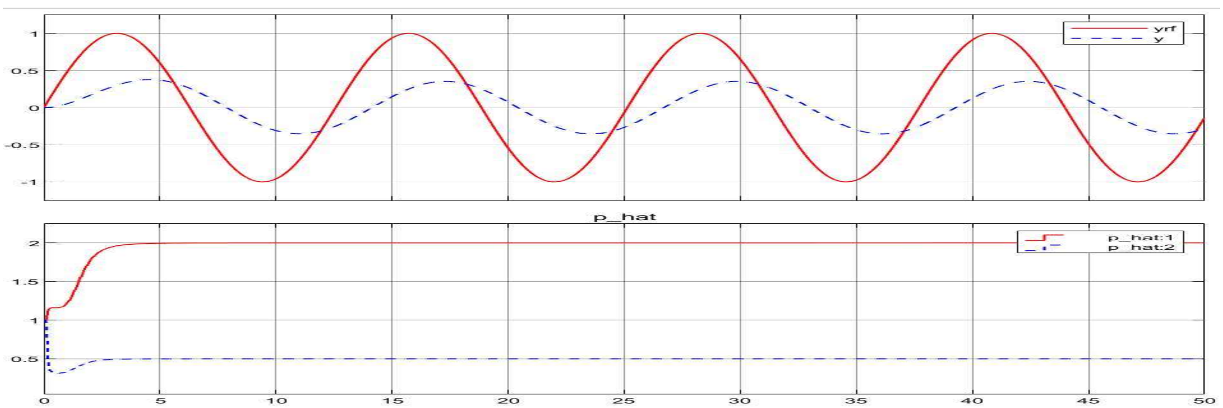


Fig5: Open loop parameter identification with **sigma2e = 0**

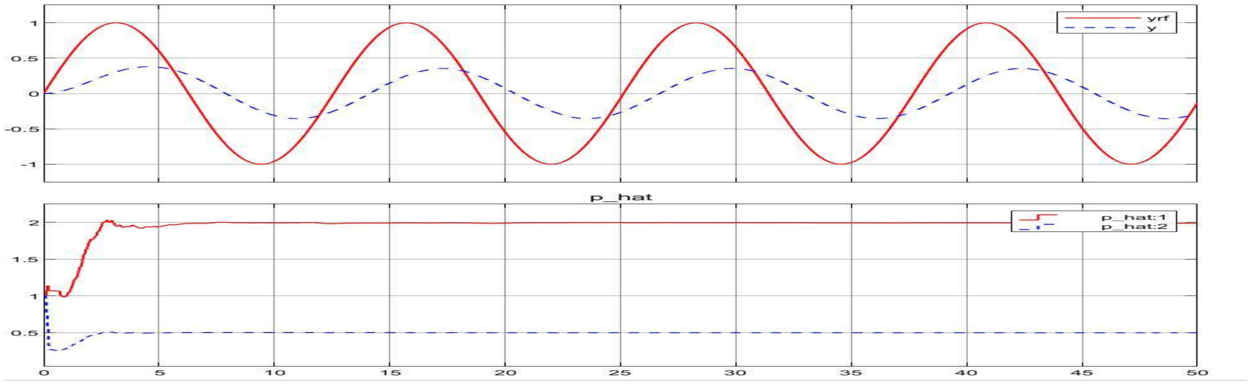


Fig6: Open loop parameter identification with **sigma2e = 0.01**

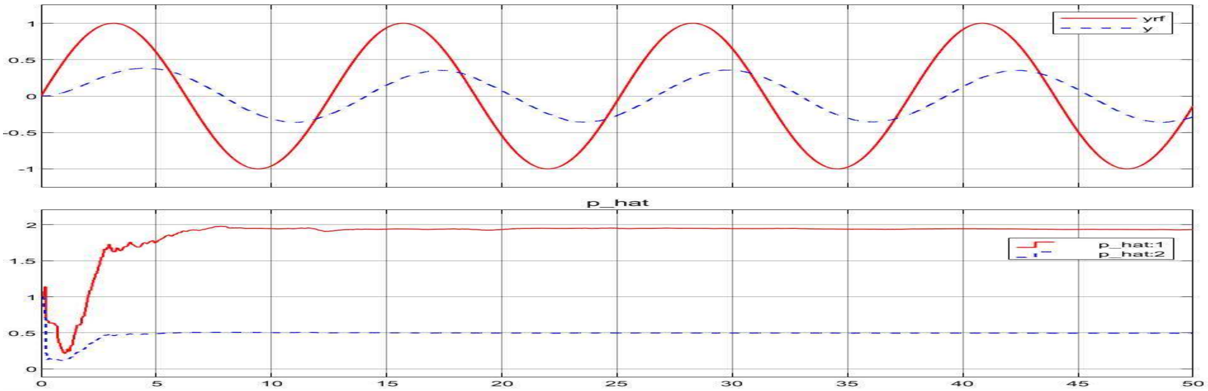


Fig6: Open loop parameter identification with **sigma2e = 0.1**

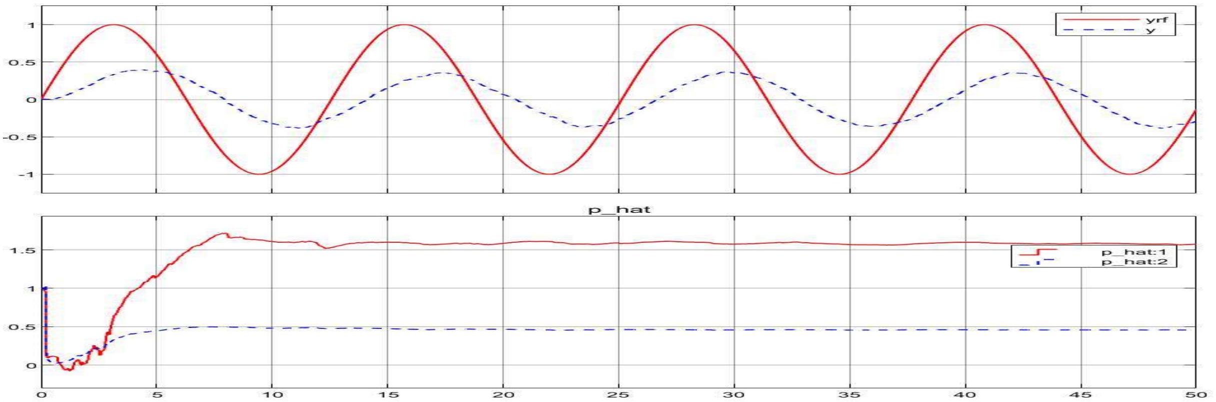


Fig7: Open loop parameter identification with **sigma2e = 1**

As we can see the figure above, the openloop plant parameter identification block able to track the real parameters in 4secs at no noise condition**(Fig1)**, whereas at **sigma2e=0.01,** the block tracked the parameters in 6secs**(Fig2)**. The worst condition at sigma2e=1, the block didn't tracked the parameters, especially the value of first parameter(**(Fig4).** But the output didn't track the reference input, because it wasn't the task of the parameter identification block, no feedback tracing.

When the value of the noise increased, the tracking capability of the plant decreases and we can conclude that noise and parameter tracking capability has **inverse** relationships.

ii. Closed loop control analysis without adaptation

- w = [5 1 1]'
- Reference signal: sinusoidal with amplitude 0.15rad/s and wr = 0.25rad/s

- simulation run time: 50sec
- Ta = 50Tc
- P = 1000*I

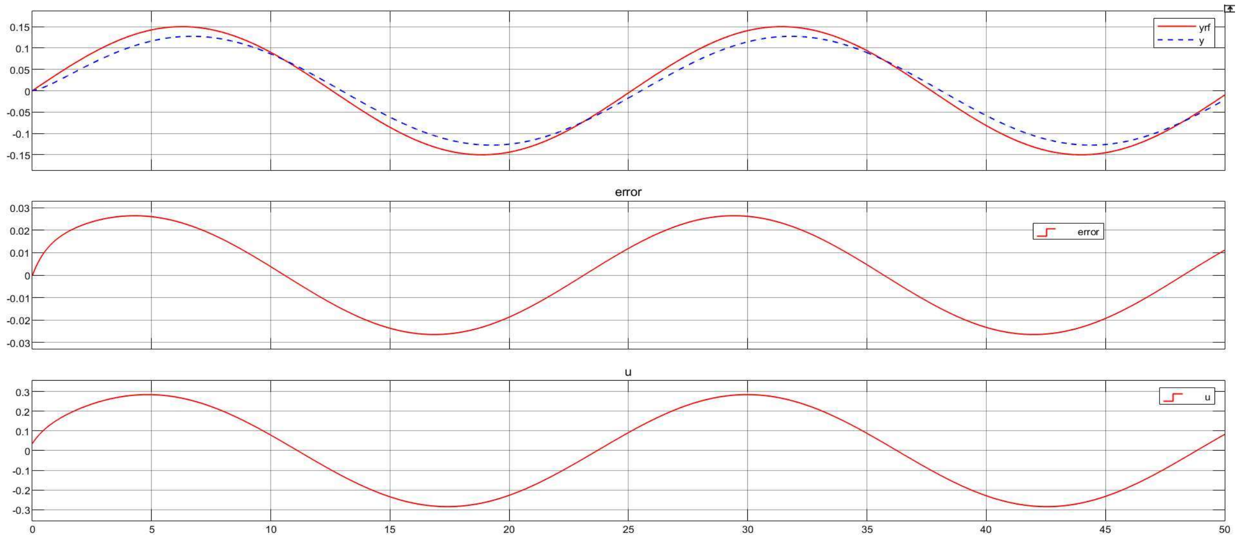
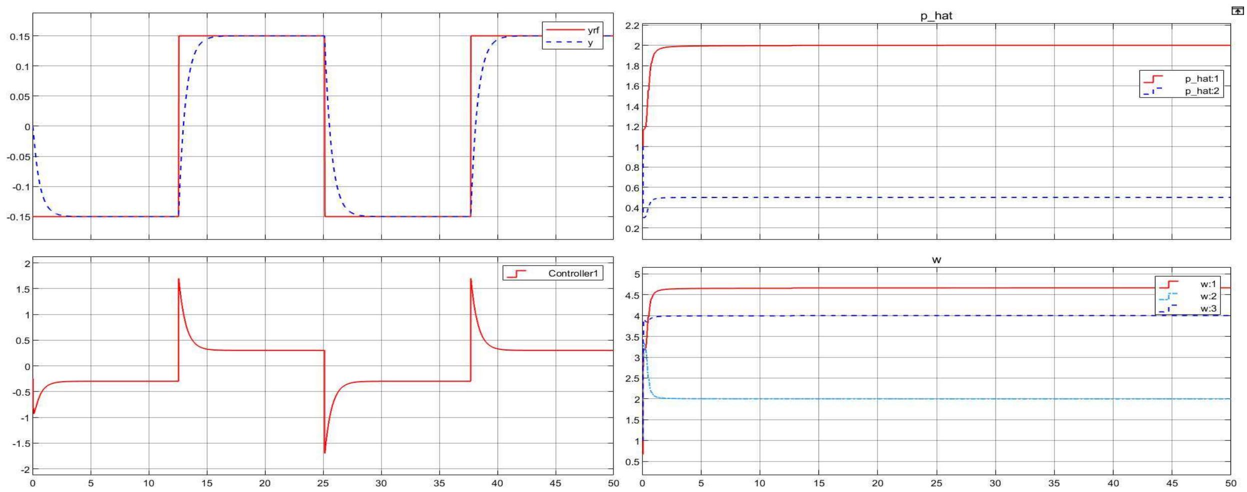


Fig8: Closed loop without adaptation at **sigma2e = 0**

The closed loop model with no adaptation didn't track the reference input perfectly. The error value is more than 1% of the expected steady state value**(Fig8)**. The plant didn't satisfy control **objective R2 & R3**. At this case, we can say the controller is not efficient and it needs further improvement. It is possible to get a better result that satisfy **objective R2&R3** by manually retunning the control gain values, but it will **not be easy** to test for each gains selected manually.

iii. Closed loop control with MIAC adaptation

- Reference signal: sinusoidal with amplitude **0.15rad/s** and **wr = 0.25rad/s**
- simulation run time: **50sec**
- Ta = 50Tc
- P = 1000*I



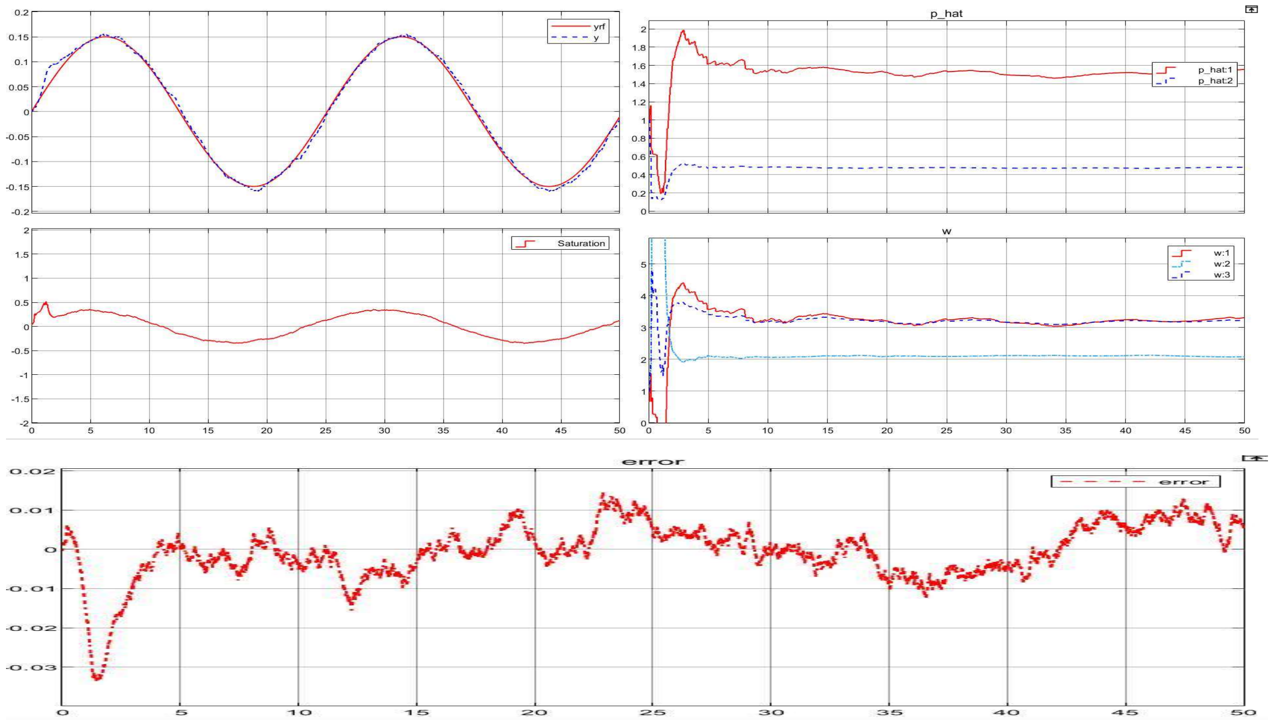


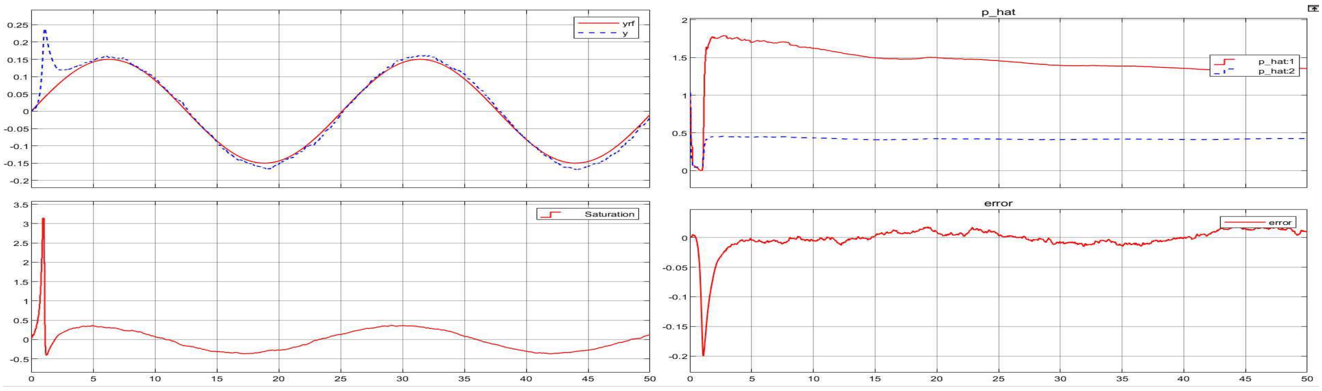
Fig12: Closed loop control with MIAC adaptation at **sigma2e = 0.1** for sinusoidal wave reference input

- Since the value of **u** is **[-0.5 0.5]**, the **saturation block has no effect** and the result is the same as results without saturation
- From the above simulation results, the square wave **satisfy the objectives R1&R3** of the adaptive control scheme set on **page 2** of this report. The steady state error **was less than 1%** of the reference input(**Fig9**) and the settling time is less than **alpha=3**. The **square wave** also showed better parameter tracking **speed and result** compared to sinusoidal wave when **noises were applied** to the plant(**Fig11**). This was due to the square wave's capability to reject sudden disturbances between the sampling transition.
- On the other hand, a sinusoidal reference input **satisfied the objectives** of the adaptive control objective. The error **was less than 1%** of the reference input when there was no noise effect(**Fig10**) and the settling time was very short(at was less than one sec). But it showed **bad parameter tracking efficiency** when noise was included compared to square wave and the objectives of the control scheme **wasn't satisfied** due to noise effect(**Fig12**).
- So, we can conclude that discontinous reference input is better under noisy conditions and continous reference input is better under zero noise conditions. The discontinuity of the reference input protected the controller from noise accumulations.

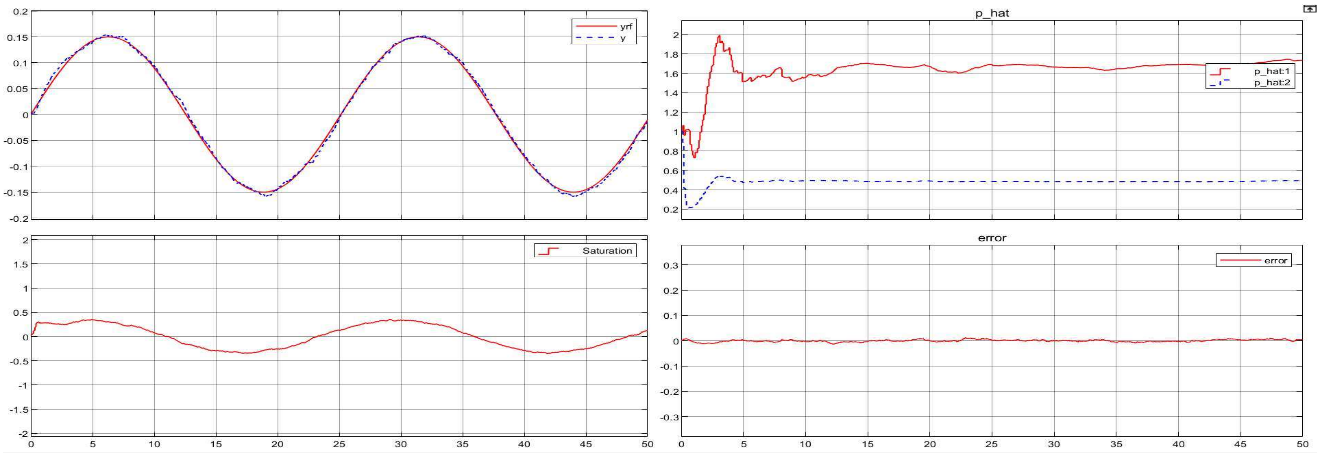
Checking the effects of some non parameteric values:

Condition 1: changing the ratio of sampling times

- Reference signal: sinusoidal with amplitude **0.15rad/s** and **wr = 0.25rad/s**
- simulation run time: **50sec**
- Ta = **10Tc** and **100Tc**
- P = 1000*I



a:



b:

Fig13: Closed loop adaptation at **sigma2e = 0.1** for sinusoidal wave reference input **a: Ta = 10Tc and b: Ta = 100Tc**

By comparing figures **Fig12, Fig13a&b**, we can say that the ratio of dwelling time (Ta) to conventional loop sampling time (Tc) highly affects the performance of the adaptive controller. As shown in **Fig13b**, the control scheme satisfied the control objectives, error less than **1%** of the reference input. This showed that the value of the reference input highly affects the plant nature and its parameter tracking capability.

As we can see on the above figures, the adaptive control scheme with higher Ta/Tc value (Ta = 100Tc) has better **overshoot, error minimization and parameter tracking** capabilities.

Condition 2: changing the reference input amplitude

- Reference signal: sinusoidal with amplitude 1**rad/s** and **wr = 0.25rad/s**
- simulation run time: **50sec**
- Ta = 50Tc
- P = 1000*I

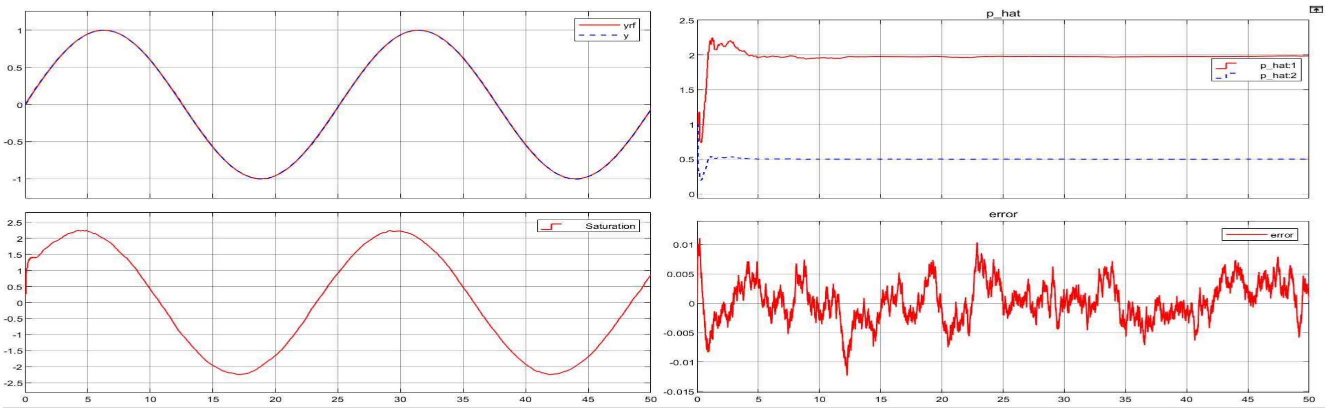


Fig14: Closed loop adaptation at **sigma2e = 0.1** for sinusoidal wave reference input **a: Ta = 50Tc and yr = 1**

Based on results of **Fig14,** increasing the value of amplitude of reference input played a positive role on error minimization, noise rejection and parameter tracking capability. As shown in the figure above, the adaptive controller satisfied objective **R2&R3** even at the worst condition (**sigma2e = 0.1**).

Condition 3: changing the covariance matrix gain 'rho'

- Reference signal: sinusoidal with amplitude **0.15rad/s** and **wr = 0.25rad/s**
- simulation run time: **50sec**
- Ta = 50Tc
- P = **10*I**

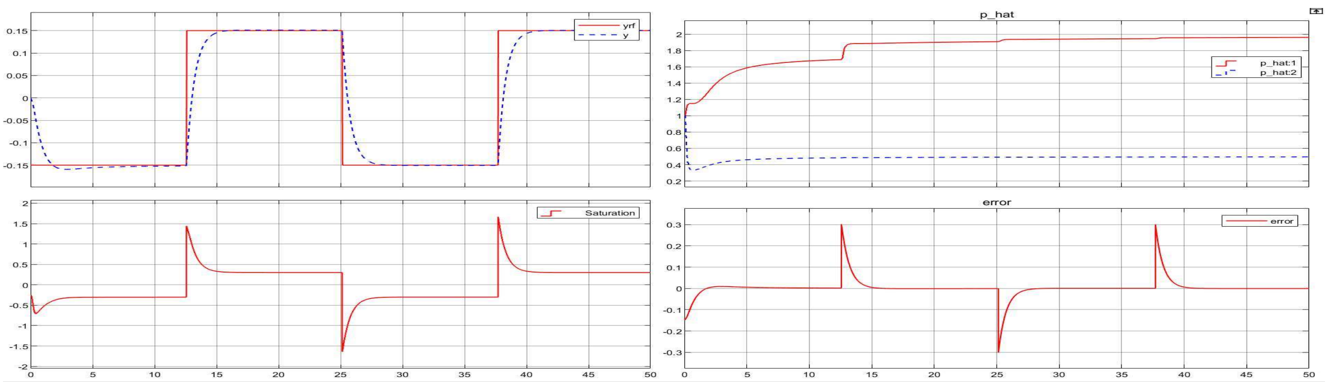


Fig15: Closed loop adaptation at **sigma2e = 0** for sinusoidal wave reference input **a: P = 10*I and yr = 0.15**

By comparing **Fig9 and Fig15**, we can see that the value of **rho** has direct relationship with the speed of parameter tracking. As the value of **rho** increased, the prameter tracking speed increased and it required a short time to reach the real parameter values.

Conclusion

Aerospace vehicles require cheap, and scalable control methods to ensure operability. Adaptive control, which doesn't require a precise model of the plant, seems excellent in meeting these requirements. In this lab activity, we explored the applicability of the **Model Identification Adaptive Control (MIAC) scheme** for non-linear input-output Aero plant dynamics, aiming on identification and parameter adaptation.

Key findings from this lab activity include:

1. MIAC, also known as indirect adaptive control (self-tunning), offers a wider applicability region for different noisy conditions. It uses arbitrary zero-pole locations and can be introduced in stages, allowing for gradual adaptation.
2. It is used for non-linear plants which have **good observability** of input-output and already **known number of parameters** without knowing their precise model.
3. The controller adapts itself online and checks the value of parameter every iteration.
4. We tested the controller by **changing parameteric and non-parameteric values** to evaluate the performance and applicability of MIAC. And the result showed that at highly noisy conditions the parameter identification block tracking capability reduced but the plant output tracked the reference acceptably.
5. Overall, the MIAC scheme satisfied the control performance objectives we stated at the initial and we can say that MIAC is very good control method.

In summary, MIAC is an excellent adaptive control scheme that can be used in Aerospace Vehicles as well as in robots, offering robustness, cost-effectiveness, and flexibility.

Appendix

Code snippets:

Initials

```
clear; clearvars;
global Ta Tc sigma2e pls P TF wr Yr alpha
Ta = 0.05; Tc = 0.001; sigma2e = 0.01;
TF = 1.5*Ta;
pls = [1 1]';
P = eye(2)*1000;
wr = 0.25; Yr = 0.15;
alpha = 3;
w = [5 1 1]';
```

Parameter Identification Block

```
function phat = miac_rls2(in)
global P pls

yf = in(1);
dyf = in(2);
uf = in(3);
phi = [-dyf uf]';
P = P - (P*(phi*phi')*P)/(1+phi'*P*phi);
kn = P*phi;
yhat = phi'*pls;
epsn = yf - yhat;
pls = pls + kn*epsn
phat = pls
```

Parameter Synthesis Block

```
function wp = miac_syn2(in)
```