ΑΣΚΗΣΗ 1

Έστω ένας εργοστασιακός φούρνος. Το αν οι αντιστάσεις του φούρνου λειτουργούν ή όχι, εξαρτάται από μια μεταβλητή C η οποία παίρνει τιμές από 0 μέχρι και 10. Με μηδέν σημαίνει ότι δεν περνάει καθόλου ρεύμα από τις αντιστάσεις του φούρνου, 10 σημαίνει ότι περνάει το μέγιστο δυνατό ρεύμα. Τοποθετείται στο εσωτερικό του φούρνου ένας αισθητήρας θερμοκρασίας. Από τις προδιαγραφές του φούρνου ξέρουμε ότι η θερμοκρασία στο εσωτερικό του, κυμαίνεται από 0 έως και 500 βαθμούς κελσίου.

1) Να οριστεί ένα κανάλι εισόδου με όνομα **Temperature** τύπου Test που θα αναπαριστά την θερμοκρασία.

- Δειγματοληψία: 2 sec
- Αριθμός διαθέσιμων δειγμάτων στην εφαρμογή: 3200

2) Να οριστεί ένα κανάλι εξόδου με όνομα **Resistor** τύπου Test που θα αναπαριστά την θερμοκρασία.

3) Να φτιαχτεί ένα sequence με όνομα **init** που θα αρχικοποιεί κάποιες μεταβλητές της εφαρμογής μας. Θα πρέπει να τρέχει κάθε φορά που ανοίγει το πρόγραμμα.

```
Resistor.AddValue(0)
```

Τρέξτε την μία τουλάχιστον φορά.

4) Να οριστεί μια μετατροπή με όνομα **Temperature_conv** που θα εφαρμόζεται στο κανάλι Temperature. Ο κώδικας που θα περιέχει θα είναι ο ακόλουθος:

```
create_temperature_channel()
```

5) Να οριστεί ένα sequence με όνομα **create_temperature_channel** με κώδικα όπως πιο κάτω:

```
If (Resistor[0]!=0)
   Private.x=7*Resistor[0]*(Log(Mean(Temperature[0,10])+1)+3)
else
   if (systime()-Resistor.Time[0]<15)
      Private.x=7*(( Resistor[1])/((systime()-
Resistor.Time[0])*3))*(Log(Mean(Temperature[0,10])+1)+3)
   else
      Private.x=0
   endif
endif
return(Private.x)
```

ΠΡΟΣΟΧΗ: Στην ίδια γραμμή

6) Αν θέσω την μεταβλητή `Resistor` ίση με 1 (Quick->Set Channel ή F2), που ισορροπεί η μεταβλητή Temperature (watch)?

7) Αν έπειτα θέσω την μεταβλητή `Resistor` ίση με 0 (Quick->Set Channel ή F2), που ισορροπεί η μεταβλητή Temperature (watch)?

8) Να φτιαχτεί ένα γραφικό περιβάλλον στην Page0 (θα την μετονομάσετε σε main_page) στο οποίο θα είναι διαθέσιμες οι ακόλουθες πληροφορίες-λειτουργίες:

Θερμοκρασία

- Να επιλέξετε ένα symbol που θα δείχνει γραφικά τον φούρνο.
- Οι τωρινές τιμές της θερμοκρασίας. Να χρησιμοποιήσετε ένα αναλογικό όργανο (gauge) και ένα ψηφιακό μετρητή.
  Θα πρέπει επίσης ο χρήστης να βλέπει ότι «κρίσιμες» τιμές θεωρούνται αυτές πάνω από 250 κοκκινίζοντας το ψηφιακό μετρητή.
- Η ιστορία των τιμών της θερμοκρασίας σε ένα δισδιάστατο γράφημα και πίνακας με τις τελευταίες 10 τιμές της .
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της θερμοκρασίας των τελευταίων 10 λεπτών.
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της θερμοκρασίας της τελευταίας 1 ώρας.
- Να προειδοποιείται με ευδιάκριτο τρόπο της επιλογής σας ο χρήστης για το αν η θερμοκρασία είναι μεγαλύτερη των 320 βαθμών (Θεωρείται επείγουσα η κατάσταση).
- Να προβλεφθεί η δυνατότητα να αποθηκεύονται οι τιμές του καναλιού σε αρχείο κειμένου. Να τοποθετήσετε ένα κουμπί που θα ξεκινά/σταματά το Logging του καναλιού. Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.

Αντίσταση

- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της Resistor (υπενθυμίζεται ότι Resistor από 0 μέχρι 10) βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα να θέτει την Resistor στην τιμή 1 και 7.
- Βρείτε κάποιο τρόπο να εξασφαλίσετε ότι η τιμή της Resistor δεν πρόκειται να γίνει <0 ούτε >10
- Να προειδοποιείται ο χρήστης αν δεν έχει αλλάξει η τιμή της Resistor για παραπάνω από 60sec (Βοήθεια: `(systime()-Resistor.time[0])`)

Γενικά

- Να γίνει ένα virtual κανάλι Temperature_history με ιστορία 5000 δειγμάτων στο οποίο θα γράφεται κάθε 1 λεπτό η τιμή Temperature[0].

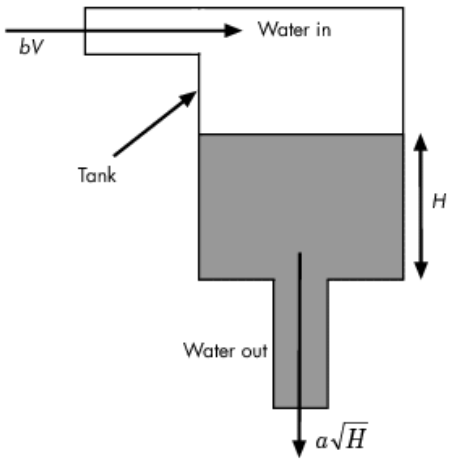
Έλεγχος

- Να οριστεί ένα καινούριο κανάλι εξόδου με όνομα **desired_Temperature**.
- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της desired_Temperature βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα να θέτει την desired_Temperature στην τιμή 0 και 200 και 320.
- Να οριστεί ένας PID ελεγκτής με τα ακόλουθα χαρακτηριστικά
    - Process Variable: **Temperature**
    - Set point: **desired_Temperature**
    - Output Channel: **Resistor**

  και αρχικά να έχω τιμές P=1, I=D=0.

- Να σχεδιαστεί ένα κουμπί με το οποίο θα ανοιγοκλείνει ο PID ελεγκτής. Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.
- Να ρυθμιστούν κατάλληλα τα P,I και D ώστε ο ελεγκτής να είναι όσο το δυνατόν πιο «γρήγορος» και με τις λιγότερες ταλαντώσεις.

ΑΣΚΗΣΗ 2



Έστω η ακόλουθη δεξαμενή όπως φαίνεται στο ακόλουθο σχήμα.

Έστω ότι είσοδος στο σύστημα είναι η **τάση *V(t)*** που εφαρμόζεται σε μια ηλεκτροβάνα που ανοιγοκλείνει την παροχή νερού στο σύστημα.

Ως έξοδος του συστήματος θεωρείται το **ύψος *H(t)*** της στάθμης του νερού**.**

Το σύστημα περιγράφεται από την ακόλουθη εξίσωση:

$$H'(k+1) = \frac{b}{a}V(k) - \frac{a}{A}\sqrt{H(k)} + H(k)$$

όπου $a=2\ \mathrm{cm}^{2.5}/\mathrm{s}$, $A=20\mathrm{cm}^2$, $b=5\ \mathrm{cm}^3/(\mathrm{s}\cdot\mathrm{V})$.

Δίνεται ότι το μέγιστο ύψος της δεξαμενής είναι 100m και ότι ο ρυθμός εισροής του νερού στη δεξαμενή είναι ανάλογος της τάσης *V(t)* όπου *V(t)* από 0 εώς 1 V.

1) Να οριστεί ένα κανάλι εισόδου με όνομα **Height** τύπου Test που θα αναπαριστά το ύψος της δεξαμενής.

- Δειγματοληψία: 1 sec
- Αριθμός διαθέσιμων δειγμάτων στην εφαρμογή: 10000

2) Να οριστεί ένα κανάλι εξόδου με όνομα **Voltage** τύπου Test που θα αναπαριστά την τάση που εφαρμόζεται στα άκρα της ηλεκτροβάνας.

3) Να φτιαχτεί ένα sequence με όνομα **init** που θα αρχικοποιεί κάποιες μεταβλητές της εφαρμογής μας. Θα πρέπει να τρέχει κάθε φορά που ανοίγει το πρόγραμμα.

```
Voltage.AddValue(0)

Height.AddValue(0)
```

Τρέξτε την μία τουλάχιστον φορά.

4) Να οριστεί μια μετατροπή με όνομα **Height_conv** που θα εφαρμόζεται στο κανάλι Height. Ο κώδικας που θα περιέχει θα είναι ο ακόλουθος:

```
create_height_channel()
```

5) Να οριστεί ένα sequence με όνομα **create_ height _channel** με κώδικα όπως πιο κάτω:

```
Var.a=2
Var.alpha=20
Var.b=5
Private.x=Var.b* Voltage[0]/Var.a -
(Var.a*Height[0]^0.5)/Var.alpha+Height[0]
```

```
If (Private.x>100)
      Private.x=100
endif
If (Private.x<0)
      Private.x=0
endif
return(Private.x)
```

6) Να εισάγετε ένα νέο κανάλι εξόδου με όνομα **overflow** (υπερχείλιση) και να βρείτε ένα τρόπο έτσι ώστε να το κανάλι αυτό να ενημερώνετε κάθε 1sec με τιμές 1 ή 0 ανάλογα με το αν το ύψος είναι μεγαλύτερο ή ίσο από 100m ή όχι (θα γίνει ποτέ > 100m??).

7) Βρείτε δοκιμάζοντας μια τιμή της μεταβλητής Voltage έτσι ώστε το ύψος της δεξαμενής να παραμένει σταθερό (όσο αυτό είναι δυνατό) (Quick->Set Channel ή F2)?

8) Να φτιαχτεί ένα γραφικό περιβάλλον στην Page0 (θα την μετονομάσετε σε main_page) στο οποίο θα είναι διαθέσιμες οι ακόλουθες πληροφορίες-λειτουργίες:

Δεξαμενή

- Να επιλέξετε ένα symbol που θα δείχνει γραφικά την δεξαμενή.
- Οι τωρινές τιμές του ύψους της δεξαμενής. Να χρησιμοποιήσετε ένα αναλογικό όργανο (led bar gauge) και ένα ψηφιακό μετρητή. Να προσαρμόσεται τα όργανα αυτά μέσα στο σύμβολο της δεξαμενής σε κατάλληλη θέση. Θα πρέπει επίσης ο χρήστης να βλέπει ότι «κρίσιμες» τιμές θεωρούνται αυτές πάνω από 80 και κάτω από 20m κοκκινίζοντας το ψηφιακό μετρητή.
- Η ιστορία των τιμών του ύψους σε ένα δισδιάστατο γράφημα και πίνακας με τις τελευταίες 7 τιμές της .
- Η μέγιστη τιμή και ο μέσος όρος των τιμών του ύψους των τελευταίων 10 λεπτών.
- Η μέγιστη τιμή και ο μέσος όρος των τιμών του ύψους των τελευταίων 2 ωρών.
- Να προειδοποιείται με ευδιάκριτο τρόπο της επιλογής σας ο χρήστης όταν το ύψος είναι μεγαλύτερο των 95 μέτρων (Θεωρείται επείγουσα η κατάσταση).
- Να προβλεφθεί η δυνατότητα να αποθηκεύονται οι τιμές του καναλιού σε αρχείο κειμένου. Να τοποθετήσετε ένα κουμπί που θα ξεκινά/σταματά το Logging του καναλιού. **Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.**

Ηλεκτροβάνα

- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της Voltage (υπενθυμίζεται ότι Voltage από 0 μέχρι 1) βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα να θέτει την Voltage στην τιμή που βρήκατε στο ερώτημα 6.

Γενικά

- Να γίνει ένα virtual κανάλι Height_history με ιστορία 5000 δειγμάτων στο οποίο θα γράφεται κάθε 1 λεπτό η τιμή Height[0].
- Να προειδοποιείται ο χρήστης αν τα τελευταία 60sec έχουμε κατάσταση υπερχείλισης περισσότερα από 50.

Έλεγχος ύψους νο1

- Να φτιαχθεί ένα sequence με όνομα **control1** το οποίο:
  - Θα τρέχει συνέχεια περιμένοντας 1sec
  - Θα ελέγχει αν το ύψος είναι μεγαλύτερο του 97 και αν είναι θα θέτει την μεταβλητή Voltage=0
  - Θα ελέγχει αν το ύψος είναι μικρότερο του 3και αν είναι θα θέτει την μεταβλητή Voltage=0.7
- Τοποθετήστε ένα κουμπί το οποίο θα ανοιγοκλείνει το sequence control1
- Αν αυτός ο αλγόριθμος ελέγχου τρέχει και δεν αλλάζουμε εμείς την μεταβλητή Voltage τι συμπεραίνεται από το 2d γράφημα του ύψους?

Έλεγχος ύψους νο2

- Να φτιαχθεί ένα sequence με όνομα **control2** το οποίο σύμφωνα με την προηγούμενη λογική θα προσπαθεί να διατηρήσει σταθερό το ύψος στην τιμή **50m** (θα σας χρειαστεί η τιμή που βρήκατε στο ερώτημα 6)
- Να φροντίσετε να μην είναι δυνατόν να τρέχουν ταυτόχρονα το control1 και control2 μαζί. (`beginseq(control1)`, `endseq(control1)`, `Local.Sequence.control1.Running`)
- Τοποθετήστε ένα κουμπί το οποίο θα ανοιγοκλείνει το sequence control2
- Αν αυτός ο αλγόριθμος ελέγχου τρέχει και δεν αλλάζουμε εμείς την μεταβλητή Voltage τι συμπεραίνεται από το 2d γράφημα του ύψους?

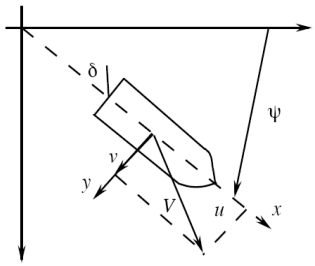
Έλεγχος ύψους νο3

- Να οριστεί ένα καινούριο κανάλι εξόδου με όνομα **desired_Height**.
- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της desired_Height βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να οριστεί ένας PID ελεγκτής με τα ακόλουθα χαρακτηριστικά
  - Process Variable: **Height**
  - Set point: **desired_Height**
  - Output Channel: **Voltage**

  και αρχικά να έχω τιμές P=1, I=D=0.

- Να σχεδιαστεί ένα κουμπί με το οποίο θα ανοιγοκλείνει ο PID ελεγκτής. Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.
- Να ρυθμιστούν κατάλληλα τα P,I και D ώστε ο ελεγκτής να είναι όσο το δυνατόν πιο «γρήγορος», με μικρότερο σφάλμα (να πηγαίνει εκεί που του λέμε το ύψος της δεξαμενής) και με τις λιγότερες ταλαντώσεις.

ΑΣΚΗΣΗ 3



Για να μειωθεί η κατανάλωση καυσίμου και η καταπόνηση των μηχανικών υλικών των πλοίων, φτιάχτηκαν συστήματα αυτόματων πιλότων που ελέγχουν την διεύθυνση πλοήγησης του πλοίου. Ένα απλό μαθηματικό μοντέλο που περιγράφει την κίνηση ενός πλοίου είναι το ακόλουθο:

$$y(k+3) = 2.73y(k+2) - 2.483y(k+1) + 0.7524y(k) - 0.00087u(k+2) - \\ -0.003432u(k+1) - 0.000839u(k)$$

όπου $\psi(t)$ η διεύθυνση του πλοίου **(γωνία πλεύσης)** και $\delta(t)$ **η γωνία του πηδάλιου** (και τα δύο σε μοίρες). Σημειώνεται ότι η γωνία πλεύσης υπολογίζεται σε ένα σταθερό σύστημα αξόνων που μετακινείται μαζί με το πλοίο όπως φαίνεται στο σχήμα, ενώ η γωνία του πηδάλιου υπολογίζεται ως προς τον άξονα του πλοίου. Και οι δύο γωνίες έχουν πρόσημο σύμφωνο με την φορά του ρολογιού. Το πλοίο έχει μια σταθερή ταχύτητα.

Δίνεται ότι τα όρια του πηδάλιου σε μοίρες είναι [-5,5]deg.

1) Να οριστεί ένα κανάλι εισόδου με όνομα **direction** τύπου Test που θα αναπαριστά την γωνία πλεύσης σε μοίρες.

- Δειγματοληψία: 1 sec
- Αριθμός διαθέσιμων δειγμάτων στην εφαρμογή: 5000

2) Να οριστεί ένα κανάλι εξόδου με όνομα **rudder** τύπου Test που θα αναπαριστά την γωνία του πηδαλίου.

3) Να φτιαχτεί ένα sequence με όνομα **init** που θα αρχικοποιεί κάποιες μεταβλητές της εφαρμογής μας. Θα πρέπει να τρέχει κάθε φορά που ανοίγει το πρόγραμμα. Να αρχικοποιηθούν όπως πρέπει οι μεταβλητές **direction και rudder**

```
direction.AddValue(0)

direction.AddValue(0)

direction.AddValue(0)

rudder.AddValue(0)

rudder.AddValue(0)

rudder.AddValue(0)
```

Τρέξτε την μία τουλάχιστον φορά.

4) Να οριστεί μια μετατροπή με όνομα **direction_conv** που θα εφαρμόζεται στο κανάλι direction. Ο κώδικας που θα περιέχει θα είναι ο ακόλουθος:

```
create_direction_channel()
```

Προσοχή σε ποια σημεία των καναλιών αντιστοιχεί πχ το y(k+2).

5) Να οριστεί ένα sequence με όνομα **create_ direction_channel** που θα ορίζει το κανάλι της γωνίας πλεύσης σύμφωνα με τον παραπάνω τύπο.

```
Private.x=2.73*direction[0] - 2.483*direction[1] +
0.7524*direction[2] - 0.00087*rudder[0] - 0.003432*rudder[1] -
0.000839*rudder[2]
```

6) Να φτιαχτεί ένα γραφικό περιβάλλον στην Page0 (θα την μετονομάσετε σε main_page) στο οποίο θα είναι διαθέσιμες οι ακόλουθες πληροφορίες-λειτουργίες:

Γωνία πλεύσης

- Να επιλέξετε ένα **symbol** που θα δείχνει γραφικά το πλοίο.
- Οι **τωρινές τιμές της γωνίας πλεύσης**. Να χρησιμοποιήσετε ένα αναλογικό όργανο (compass) και ένα ψηφιακό μετρητή.
  Η **ιστορία των τιμών της γωνίας πλεύσης** σε ένα δισδιάστατο γράφημα και πίνακας με τις τελευταίες 7 τιμές της .
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της γωνίας πλεύσης των τελευταίων 10 λεπτών.
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της γωνίας πλεύσης των τελευταίων 2 ωρών.
- Να προειδοποιείται με ευδιάκριτο τρόπο της επιλογής σας ο χρήστης όταν η γωνία πλεύσης γίνεται μικρότερη του -10 ή μεγαλύτερη του 10.
- Να προβλεφθεί η δυνατότητα να αποθηκεύονται οι τιμές του καναλιού σε αρχείο κειμένου. Να τοποθετήσετε ένα κουμπί που θα ξεκινά/σταματά το Logging του καναλιού. **Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.**

Πηδάλιο

- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της rudder (υπενθυμίζεται ότι rudder από -5 μέχρι 5 μοίρες) βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα να θέτει την rudder στην τιμή 0.

Γενικά

- Να γίνει ένα virtual κανάλι direction_history με ιστορία 10000 δειγμάτων στο οποίο θα γράφεται κάθε 1 λεπτό η τιμή direction[0].

Έλεγχος διεύθυνσης νο1 – Σταθερή γωνία πλεύσης στο 0.

- Να φτιαχθεί ένα sequence με όνομα **control1** το οποίο:
  - Θα τρέχει συνέχεια περιμένοντας 1sec

- Θα ελέγχει αν το η γωνία πλεύσης είναι μεγαλύτερη του 2 και θα θέτει σωστά την μεταβλητή rudder
- Θα ελέγχει αν το η γωνία πλεύσης είναι μικρότερη του -2 και θα θέτει σωστά την μεταβλητή rudder

- Τοποθετήστε ένα κουμπί το οποίο θα ανοιγοκλείνει το sequence control1
- Αν αυτός ο αλγόριθμος ελέγχου τρέχει και δεν αλλάζουμε εμείς την μεταβλητή rudder τι συμπεραίνεται από το 2d γράφημα της γωνίας πλεύσης?

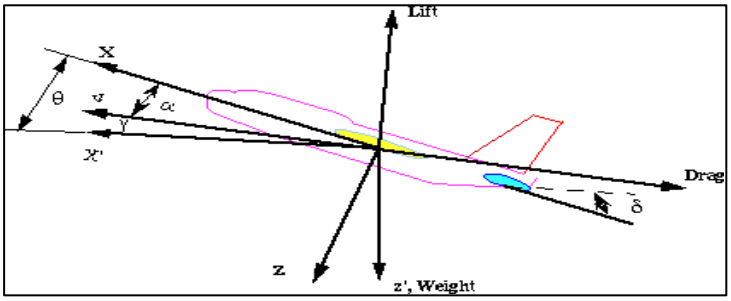
Έλεγχος διεύθυνσης vo2 – Επιθυμητή γωνία πλεύσης.

- Να οριστεί ένα καινούριο κανάλι εξόδου με όνομα **desired_direction** από -15 εώς 15 μοίρες.
- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της desired_ direction βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να οριστεί ένας PID ελεγκτής με τα ακόλουθα χαρακτηριστικά
  - Process Variable: **direction**
  - Set point: **desired_ direction**
  - Output Channel: **rudder**

  και αρχικά να έχω τιμές P=-0.5, I=D=0.

- Να σχεδιαστεί ένα κουμπί με το οποίο θα ανοιγοκλείνει ο PID ελεγκτής. Να βρεθεί ένας τρόπος ώστε το κουμπί να γράφει ή start ή stop αναλόγως.
- Να ρυθμιστούν κατάλληλα τα P,I και D ώστε ο ελεγκτής να είναι όσο το δυνατόν πιο «γρήγορος», με μικρότερο σφάλμα (να πηγαίνει στη σωστή διεύθυνση το πλοίο) και με τις λιγότερες ταλαντώσεις.

ΑΣΚΗΣΗ 4η



Έστω ένα **αεροπλάνο σε πτήση** όπως στο σχήμα. Η μοντελοποίηση του αεροπλάνου σε ώρα πτήσης είναι ένα πολύπλοκο πρόβλημα που καταλήγει σε έξι μη γραμμικές διαφορικές εξισώσεις. Οι βασικές δυνάμεις που επιδρούν στο αεροσκάφος φαίνονται στο σχήμα. Υποθέτουμε ότι το αεροπλάνο πετάει σε σταθερό ύψος, με σταθερή ταχύτητα. Επιπλέον για ευκολία ας υποθέσουμε ότι μια αλλαγή στην κάθετη γωνία πτήσης δεν επιφέρει αλλαγή στην ταχύτητα. **Στόχος είναι ο έλεγχος της κάθετης γωνίας πτήσης του αεροπλάνου θ μέσω της γωνίας των πίσω πτερυγίων δ.** Και οι δύο γωνίες μετριούνται σε **rad**. Το σύστημα το οποίο αντιπροσωπεύει ένα μοντέλο της **Boeing** δίνεται από την ακόλουθη εξίσωση διαφορών:

$$\theta(k+3) = 1.874\theta(k+2) - 1.352\theta(k+1) + 0.477\theta(k) + 0.44\delta(k+2) - 0.01\delta(k+1) - 0.31\delta(k)$$

Δίνεται ότι τα όρια της γωνίας των πίσω πτερυγίων σε ακτίνια είναι [-0.15, 0.15]rad.

1. Να οριστούν τα κατάλληλα κανάλια εισόδου και εξόδου και όπου αυτό χρειάζεται να μπει
   - Δειγματοληψία: 1 sec
   - Αριθμός διαθέσιμων δειγμάτων στην εφαρμογή: 4000
2. Να φτιαχθεί κώδικας που θα «γεμίζει» το κανάλι που θα αντιστοιχεί στην γωνία πτήσης σύμφωνα με τον παραπάνω τύπο.
3. Να φτιαχτεί ένα γραφικό περιβάλλον στην Page0 (θα την μετονομάσετε σε main_page) στο οποίο θα είναι διαθέσιμες οι ακόλουθες πληροφορίες-λειτουργίες:

Κάθετη γωνία πτήσης του αεροπλάνου

- Να επιλέξετε ένα **symbol** που θα δείχνει γραφικά το αεροπλάνο.
- Οι **τωρινές τιμές της γωνίας πτήσης**. Να χρησιμοποιήσετε ένα αναλογικό όργανο (όποιο θέλετε/ταιριάζει) και ένα ψηφιακό μετρητή.
  Η **ιστορία των τιμών της γωνίας πτήσης** σε ένα δισδιάστατο γράφημα και πίνακας με τις τελευταίες 6 τιμές της .
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της γωνίας πτήσης των τελευταίων 5 λεπτών.
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της γωνίας πτήσης των τελευταίων 2 ωρών.
- Να προειδοποιείται με ευδιάκριτο τρόπο της επιλογής σας ο χρήστης όταν η γωνία πτήσης γίνεται μικρότερη του -0.4 ή μεγαλύτερη του 0.4.

- Να προβλεφθεί η δυνατότητα να αποθηκεύονται οι τιμές του καναλιού σε αρχείο κειμένου. Να τοποθετήσετε ένα κουμπί που θα ξεκινά/σταματά το Logging του καναλιού.

Πίσω πτερύγια

- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) την τιμή της γωνίας των πίσω πτερυγίων βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα να την θέτει στην τιμή 0.

Γενικά

- Να γίνει ένα virtual κανάλι με ιστορία 8000 δειγμάτων στο οποίο θα γράφεται κάθε 1 λεπτό η τιμή της γωνίας πτήσης.

Έλεγχος διεύθυνσης vo1 – Σταθερή γωνία πτήσης στο 0.

- Να φτιαχθεί ένα sequence με όνομα **control1** το οποίο:
  - Θα τρέχει συνέχεια περιμένοντας 1sec
  - Θα ελέγχει αν το η γωνία πτήσης είναι μεγαλύτερη του 0.1 και θα θέτει σωστά την γωνία των πίσω πτερυγίων
  - Θα ελέγχει αν το η γωνία πτήσης είναι μικρότερη του -0.1 και θα θέτει σωστά την γωνία των πίσω πτερυγίων
- Τοποθετήστε ένα κουμπί το οποίο θα ανοιγοκλείνει το sequence control1

Έλεγχος διεύθυνσης vo2 – Επιθυμητή γωνία πτήσης.

- Να οριστεί ένα καινούριο κανάλι εξόδου που θα αναπαριστά την επιθυμητή γωνία πτήσης από -0.3 εώς 0.3 rad.
- Να μπορεί ο χρήστης να την μεταβάλει με γραφικό τρόπο (knob) βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να οριστεί ένας PID ελεγκτής που θα προσπαθεί να θέσει την γωνία πτήσης ίση με την επιθυμητή γωνία.
- Να ρυθμιστούν κατάλληλα τα P,I και D ώστε ο ελεγκτής να είναι όσο το δυνατόν πιο «γρήγορος», με μικρότερο σφάλμα (να πηγαίνει στη σωστή γωνία το αεροπλάνο) και με τις λιγότερες ταλαντώσεις.
- Να σχεδιαστεί ένα κουμπί με το οποίο θα ανοιγοκλείνει ο PID ελεγκτής εξασφαλίζοντας ότι το **control1** θα είναι ήδη σταματημένο.

ΑΣΚΗΣΗ 4η

Έστω ένα αυτοκίνητο σε ευθεία κίνηση. Ας θεωρήσουμε για λόγους απλότητας ότι το αυτοκίνητο δεν έχει φρένα. Η ταχύτητα του αυτοκινήτου v(k) την χρονική στιγμή k, εξαρτάται άμεσα από μια μεταβλητή p(k) που είναι ανάλογη με το πόσο πατημένο είναι το γκάζι. Το σύστημα περιγράφεται από την ακόλουθη εξίσωση:

$$v(k+1) = 0.951229\mathrm{v(k)}+\mathrm{p(k)}$$

Θεωρούμε ότι στο συγκεκριμένο αυτοκίνητο η μέγιστη ταχύτητα είναι 220km/h και ότι η μεταβλητή p(k) κυμαίνεται στο διάστημα [0,10].

1. Να οριστούν τα κατάλληλα κανάλια εισόδου και εξόδου και όπου αυτό χρειάζεται να μπει
    - Δειγματοληψία: 1 sec
    - Αριθμός διαθέσιμων δειγμάτων στην εφαρμογή: 5000
2. Να φτιαχθεί κώδικας που θα «γεμίζει» το κανάλι που θα αντιστοιχεί στην ταχύτητα του αυτοκινήτου σύμφωνα με τον παραπάνω τύπο.
3. Να φτιαχτεί ένα γραφικό περιβάλλον στην Page0 (θα την μετονομάσετε σε main_page) στο οποίο θα είναι διαθέσιμες οι ακόλουθες πληροφορίες-λειτουργίες:

Ταχύτητα του αυτοκινήτου

- Να επιλέξετε ένα **symbol** που θα δείχνει γραφικά το αυτοκίνητο.
- Οι **τωρινές τιμές της ταχύτητας του αυτοκινήτου**. Να χρησιμοποιήσετε ένα αναλογικό όργανο (όποιο θέλετε/ταιριάζει) και ένα ψηφιακό μετρητή. Να φαίνονται ευδιάκριτα τα όρια 50km/h, 120km/h και 140km/h.
  Η **ιστορία των τιμών της ταχύτητας του αυτοκινήτου** σε ένα δισδιάστατο γράφημα και πίνακας με τις τελευταίες 6 τιμές της .
- Η μέγιστη τιμή και ο μέσος όρος των τιμών της ταχύτητας του αυτοκινήτου των τελευταίων 1 λεπτών.
- Να προειδοποιείται με ευδιάκριτο τρόπο της επιλογής σας ο χρήστης όταν η ταχύτητα γίνεται μεγαλύτερη του 140.
- Να προβλεφθεί η δυνατότητα να αποθηκεύονται οι τιμές του καναλιού σε αρχείο κειμένου. Να τοποθετήσετε ένα κουμπί που θα ξεκινά/σταματά το Logging του καναλιού.

Γκάζι

- Να μπορεί ο χρήστης να μεταβάλει με γραφικό τρόπο (knob) το πόσο πατημένο είναι το γκάζι βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να μπορεί ο χρήστης γρήγορα θέτει στο γκάζι στην τιμή 0.

Γενικά

- Να γίνει ένα virtual κανάλι με ιστορία 5000 δειγμάτων στο οποίο θα γράφεται κάθε 1 λεπτό η τιμή της ταχύτητας.

Έλεγχος ταχύτητας vo1 – Σταθερή ταχύτητα (cruise control) περίπου στα 100km/h.

- Να φτιαχθεί ένα sequence με όνομα **control1** το οποίο:
  - Θα τρέχει συνέχεια περιμένοντας 1sec
  - Θα ελέγχει αν η ταχύτητα είναι μεγαλύτερη του 95 και θα θέτει σωστά το γκάζι
  - Θα ελέγχει αν η ταχύτητα είναι μικρότερη του 105 και θα θέτει σωστά το γκάζι
- Τοποθετήστε ένα κουμπί το οποίο θα ανοιγοκλείνει το sequence control1

Έλεγχος ταχύτητας vo1 – Επιθυμητή ταχύτητα (cruise control)

- Να οριστεί ένα καινούριο κανάλι εξόδου που θα αναπαριστά την επιθυμητή ταχύτητα από 0 εώς 220km/h.
- Να μπορεί ο χρήστης να την μεταβάλει με γραφικό τρόπο (knob) βλέποντας ταυτόχρονα και ψηφιακά την τιμή της.
- Να οριστεί ένας PID ελεγκτής που θα προσπαθεί να θέσει την πραγματική ταχύτητα ίση με την επιθυμητή ταχύτητα.
- Να ρυθμιστούν κατάλληλα τα P,I και D ώστε ο ελεγκτής να είναι όσο το δυνατόν πιο «γρήγορος», με μικρότερο σφάλμα και με τις λιγότερες ταλαντώσεις.