

PID Ανατροφοδότηση κλειστού βρόχου

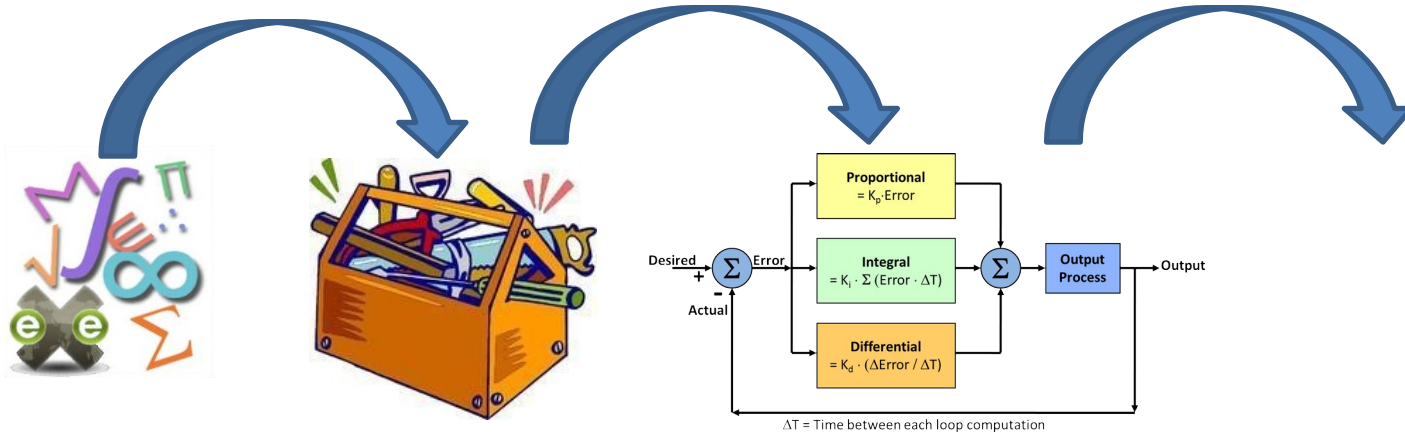
Θεωρία ελέγχου για διασκέδαση και όφελος

Εργαστήριο Προχωρημένων Θεμάτων

Scott Gray

Απόδοση: 8^ο Γυμνάσιο/7^ο ΓΕΛ Καλαμαριάς

Περιεχόμενα

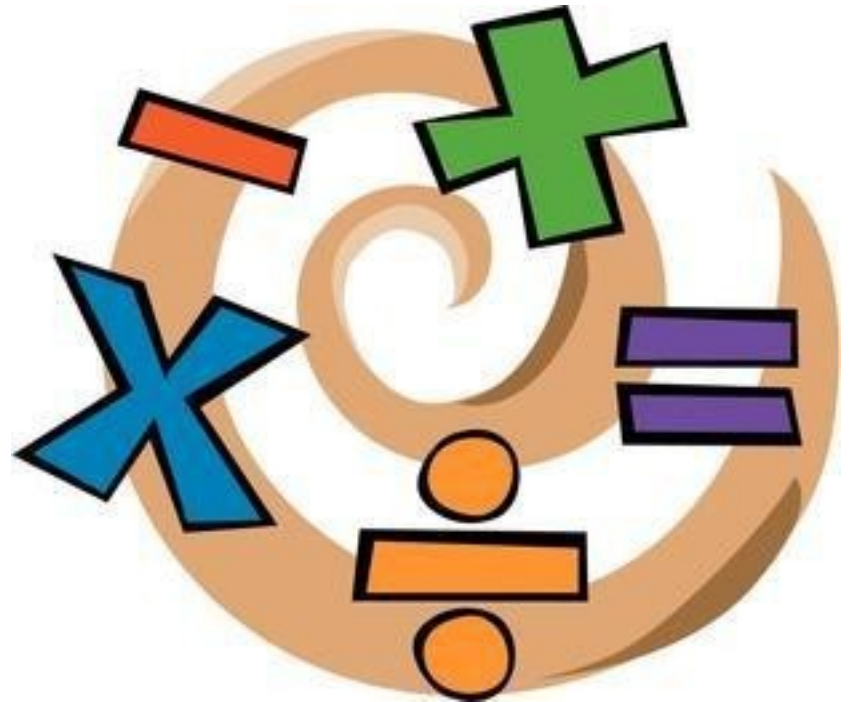


1. Μαθηματικό λεξιλόγιο, σύμβολα και μερικά μαθηματικά εργαλεία για την εργαλειοθήκη της γνώσης μας
2. Θεωρία και εφαρμογή των κλειστών βρόχων ανατροφοδότησης PID για την εργαλειοθήκη των δεξιοτήτων μας
3. Εφαρμογή βρόχου ελέγχου PID για ρομπότ που ακολουθούν γραμμή

Λίγο Λεξιλόγιο Μαθηματικών

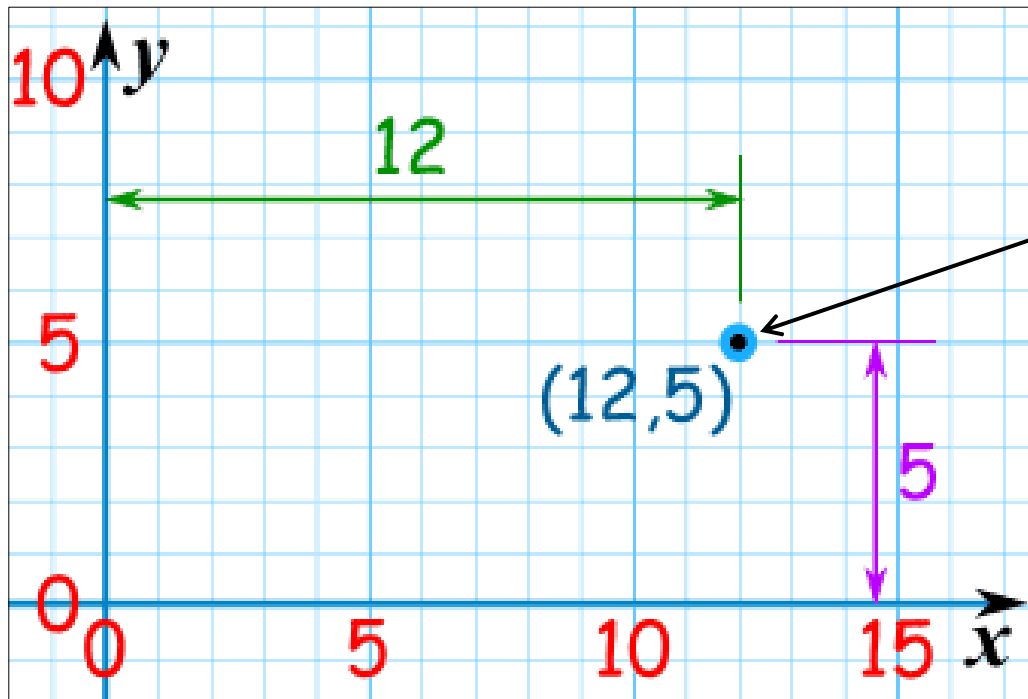
Για να κατανοήσετε τα μαθηματικά που θα ακολουθήσουν, είναι σημαντικό να διαβάσετε και/ή να μάθετε μερικούς μαθηματικούς όρους και μαθηματικά σύμβολα:

1. Σημείο
2. Άπειρο - ∞
3. Γραμμή
4. Γραμμή εφαπτομένης
5. Μεταβλητή (Variable)
6. Συνάρτηση (Function)
7. Αλλαγή (Διαφορά) - Δ
8. Κλίση
9. Σειρά (Άθροισμα) - Σ



Τι είναι το σημείο;

Ένα σημείο είναι μια ακριβής τοποθεσία. Δεν είναι ένα πράγμα, αλλά ένας τόπος. Δεν έχει μέγεθος ή διαστάσεις, απλώς χρησιμοποιούμε μια τελεία για να αναπαραστήσουμε πού βρίσκεται ένα σημείο.



Αυτό το σημείο $(12, 5)$ είναι 12 μονάδες κατά μήκος και 5 μονάδες πάνω.

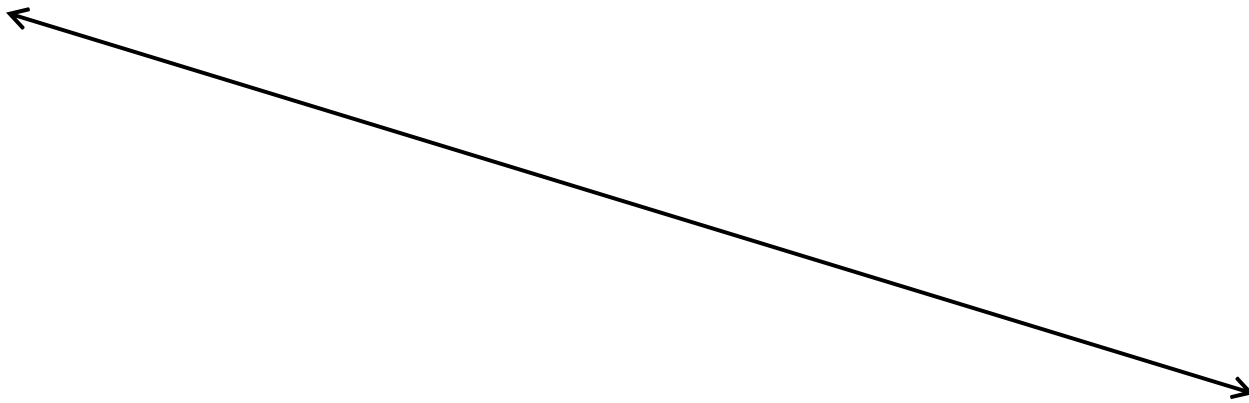
Η Σοφία του Απείρου

Το άπειρο είναι η ιδέα ότι κάτι δεν έχει τέλος. Αν κάποιος έχει διαβάσει κάθε βιβλίο για τις πυραμίδες, θα μπορούσατε να πείτε ότι έχει άπειρες γνώσεις για τις πυραμίδες (αυτό φυσικά θα ήταν υπερβολή του «άπειρου»). Σίγουρα θα σταματήσει να μιλά για αυτές κάποια στιγμή, σωστά;

Σύμβολο του Απείρου: 

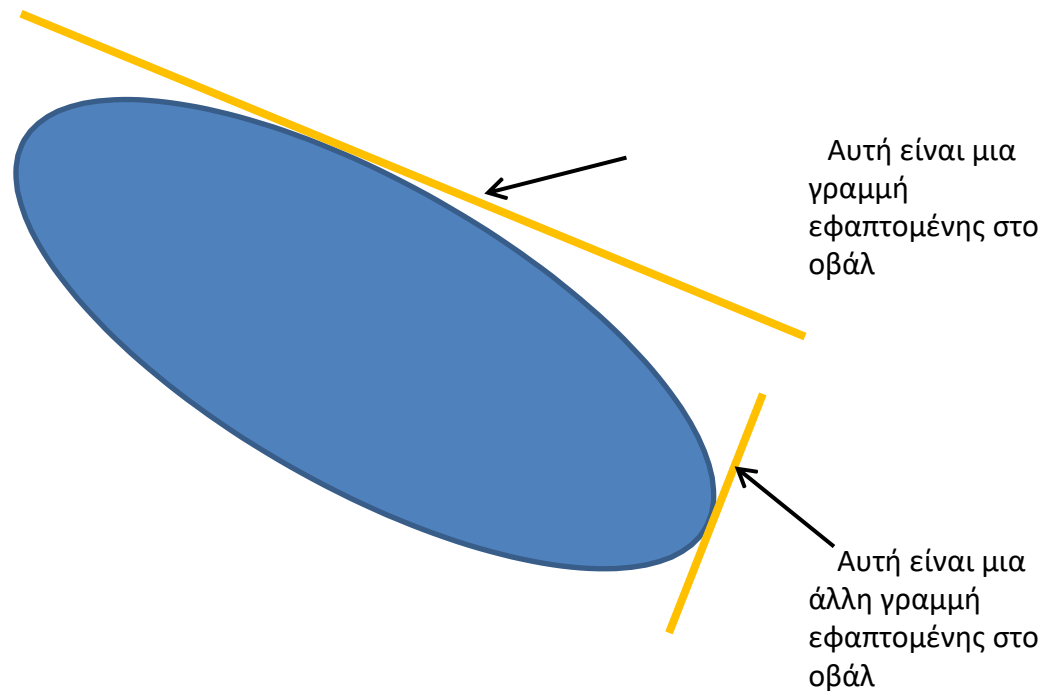
Περπατάς στη γραμμή;

Μια γραμμή είναι μια ευθεία, μια σειρά μιας διάστασης από άπειρα σημεία. Εάν σχεδιάσετε μια γραμμή με ένα μολύβι, απλώς αντιπροσωπεύει πού είναι η γραμμή, γιατί αν κοιτάξετε τη γραμμή στο μικροσκόπιο θα έδειχνε μια γραμμή με μεγάλο πλάτος!



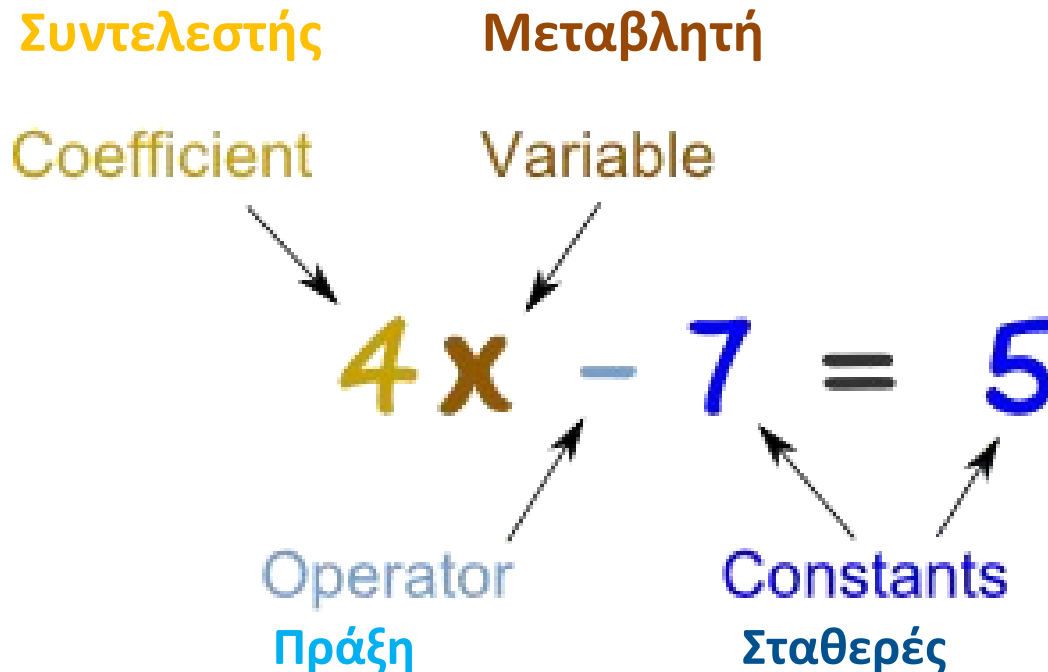
Εφαπτομενική σκέψη

Μία **γραμμή εφαπτομένης** αγγίζει μια καμπύλη σε ένα μόνο σημείο (τοποθεσία). Η «ακτίνα» αυτής της καμπύλης είναι πάντα κάθετη (ορθή γωνία, 90 μοίρες) στην εφαπτομένη.



Οι μεταβλητές κρατούν την τιμή

Μια μεταβλητή είναι απλώς ένα σύμβολο, ή γράμμα, ή λέξη, ή ακόμα και μια φράση (ThisIsALongVariableName) που μπορεί να περιέχει μια τιμή (έναν ακέραιο, έναν πραγματικό αριθμό ή ακόμα και ένα χρώμα!).



Συναρτήσεις (Functions): Μηχανές Μαθηματικών!

Μια συνάρτηση συσχετίζει κάποια είσοδο ή εισόδους με κάποια έξοδο. Μια συνάρτηση είναι σαν ένα μηχάνημα που στροβιλίζεται στην είσοδο και παράγει μια έξοδο.

The diagram shows the mathematical expression $f(x) = x^2$. The function name 'f' is in blue, the input 'x' is in purple, and the output 'x^2' is in orange. A blue arrow points from the label 'function name' to 'f'. A purple arrow points from the label 'input' to 'x'. An orange bracket under 'x^2' is labeled 'what to output'.

Μια συνάρτηση παίρνει ένα όνομα, όπως "f", ή "g", ή "PIG" ή "AREA". Μια συνάρτηση απλώς κάνει κάποια ενέργεια στην είσοδο για να κάνει μια έξοδο.

Η αλλαγή θα σου κάνει καλό!

Ένα χρήσιμο σύμβολο στα μαθηματικά και στους μηχανικούς είναι το:

Σύμβολο Δέλτα: Δ

Αντιπροσωπεύει την αλλαγή στην τιμή (ή τη διαφορά στην τιμή).

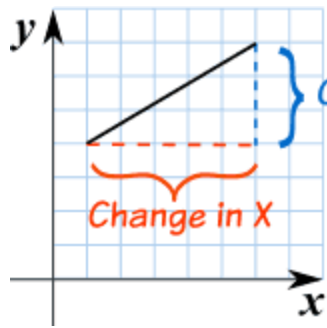
$\Delta =$ αλλαγή στην τιμή

Εάν η τιμή ενός αισθητήρα φωτός αλλάξει από 75 σε 50 από τη μια φορά που παίρνουμε ένα δείγμα και στην άλλη, λέμε ότι έχει «Δέλτα», ή $\Delta = -25$.

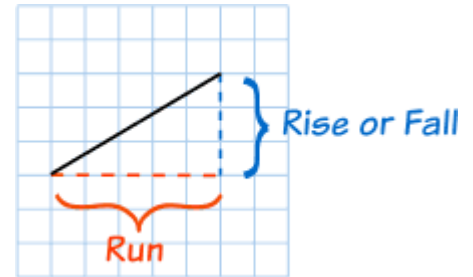
Κλίση

Κλίση της γραμμής:

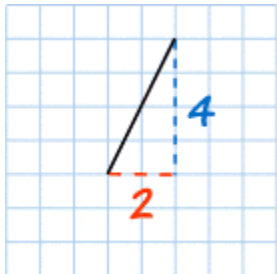
Μπορούμε να βρούμε την κλίση μιας οποιασδήποτε γραμμής σε μια γραφική παράσταση



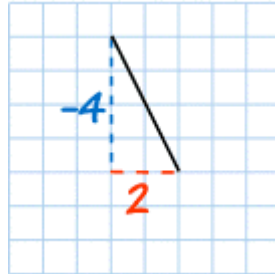
$$\text{Κλίση} = \frac{\text{Αλλαγή στο } Y}{\text{Αλλαγή στο } X} = \frac{\Delta Y}{\Delta X} = m =$$



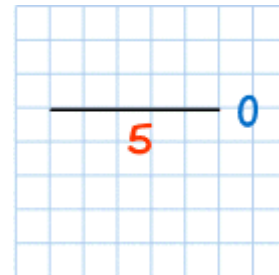
Παραδείγματα:



$$m = \frac{\Delta Y}{\Delta X} = \frac{4}{2} = 2$$



$$m = \frac{-4}{2} = -2$$



$$m = \frac{0}{5} = 0$$

Flat!

Συνοψίζοντας, αθροίζοντας...

Ένα άλλο χρήσιμο σύμβολο στα μαθηματικά και στους μηχανικούς είναι

το σύμβολο της Σειράς (Άθροισμα): Σ

Απεικονίζει να συγκεντρώνεις μαζί, να προσθέτεις μια πληθώρα τιμών.

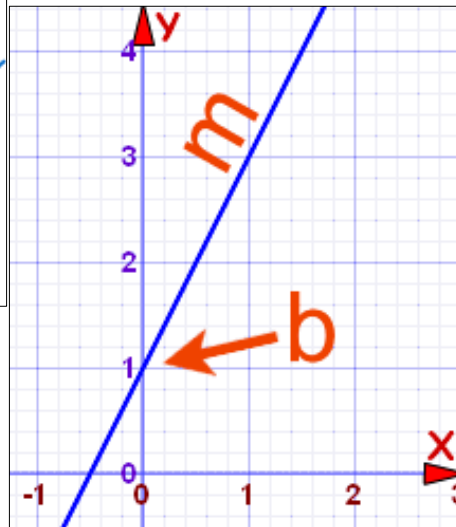
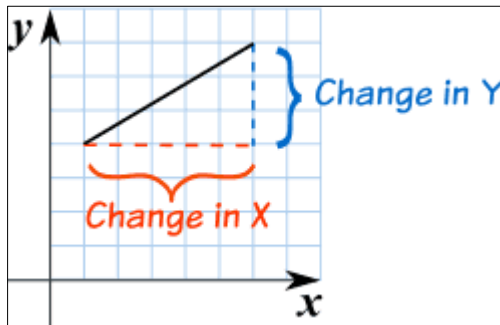
$\Sigma = \text{άθροισμα των τιμών}$

Για παράδειγμα, το Σ όλων των ακεραίων από 1 στο 9 είναι $\Sigma = 1+2+3+4+5+6+7+8+9 = 45$.

Άλγεβρα που μας νοιάζει!

Εξίσωση γραμμής:

Χρησιμοποιώντας μεταβλητές, μπορούμε να περιγράψουμε οποιαδήποτε γραμμή σε ένα γράφημα:



$$\begin{aligned} \text{Κλίση} &= \frac{\text{Αλλαγή στο } Y}{\text{Αλλαγή στο } X} \\ &= \frac{\Delta Y}{\Delta X} = m \end{aligned}$$

$$y = mx + b$$

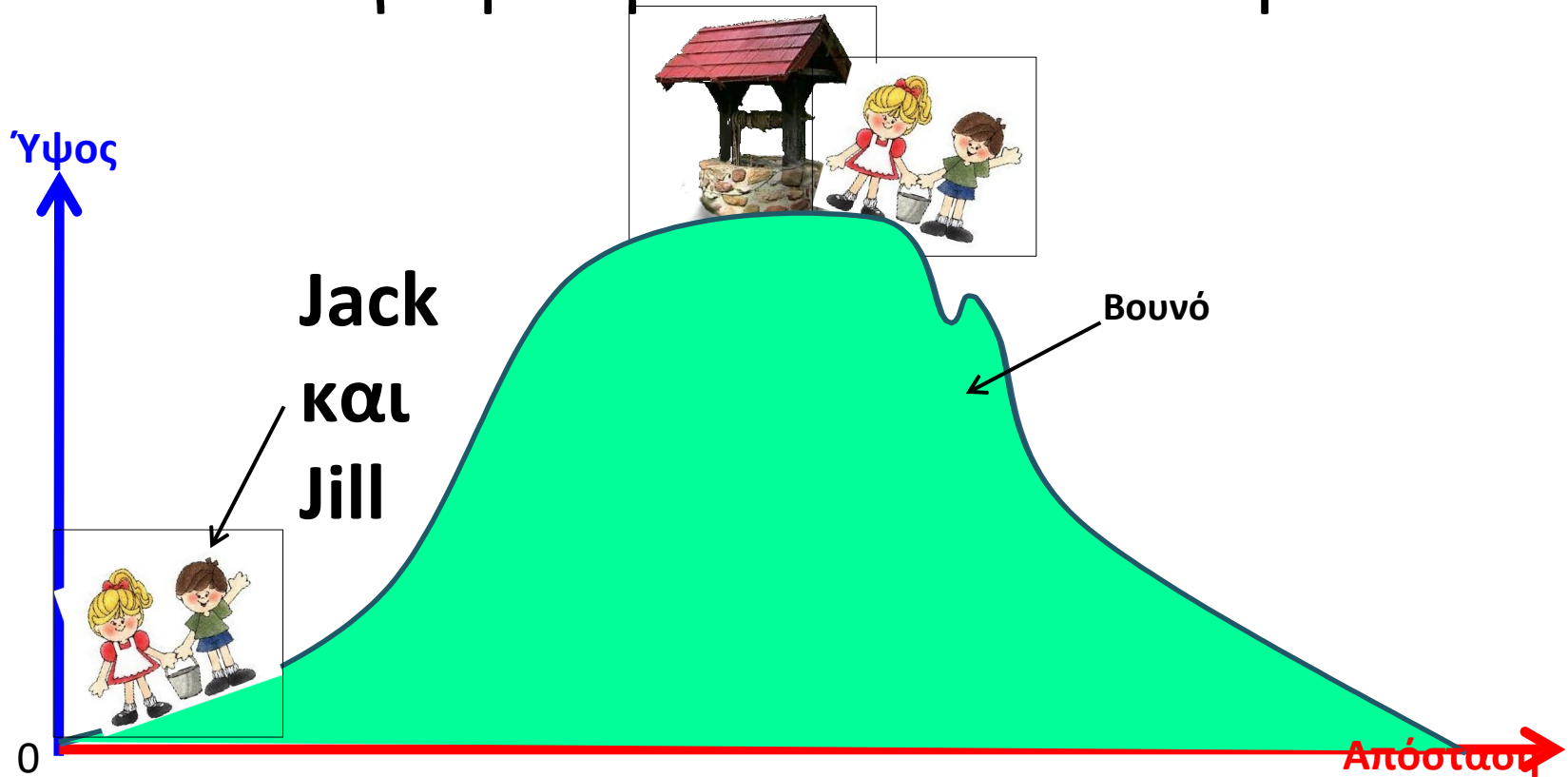
y = πόσο μακριά προς τα πάνω

x = πόσο μακριά οριζόντια

m = Κλίση (πόσο απότομη η γραμμή είναι)

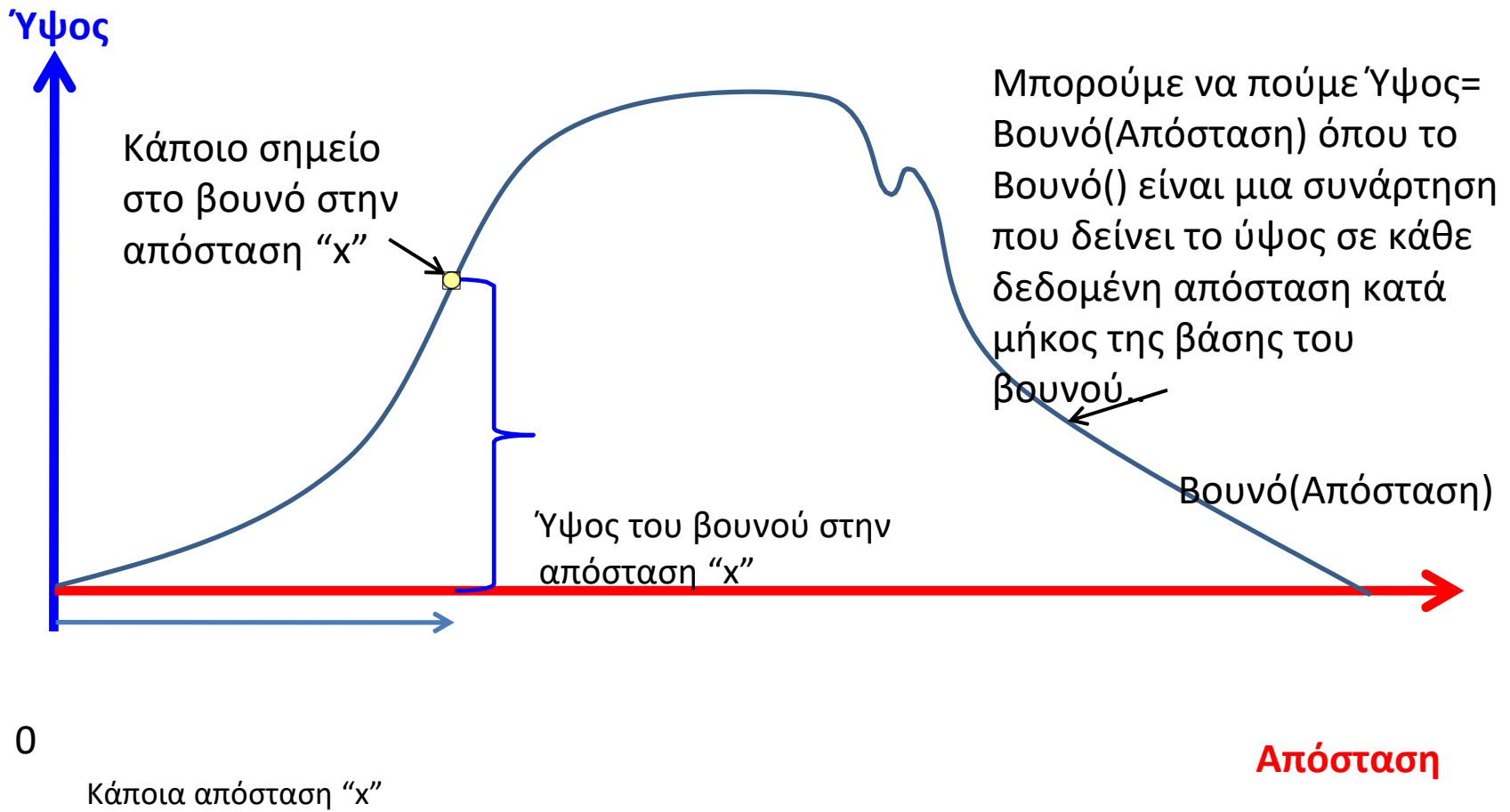
b = το Y σημείο τομής (όπου η γραμμή τέμνει τον Y άξονα)

Υπολογισμοί με τον Jack και την Jill



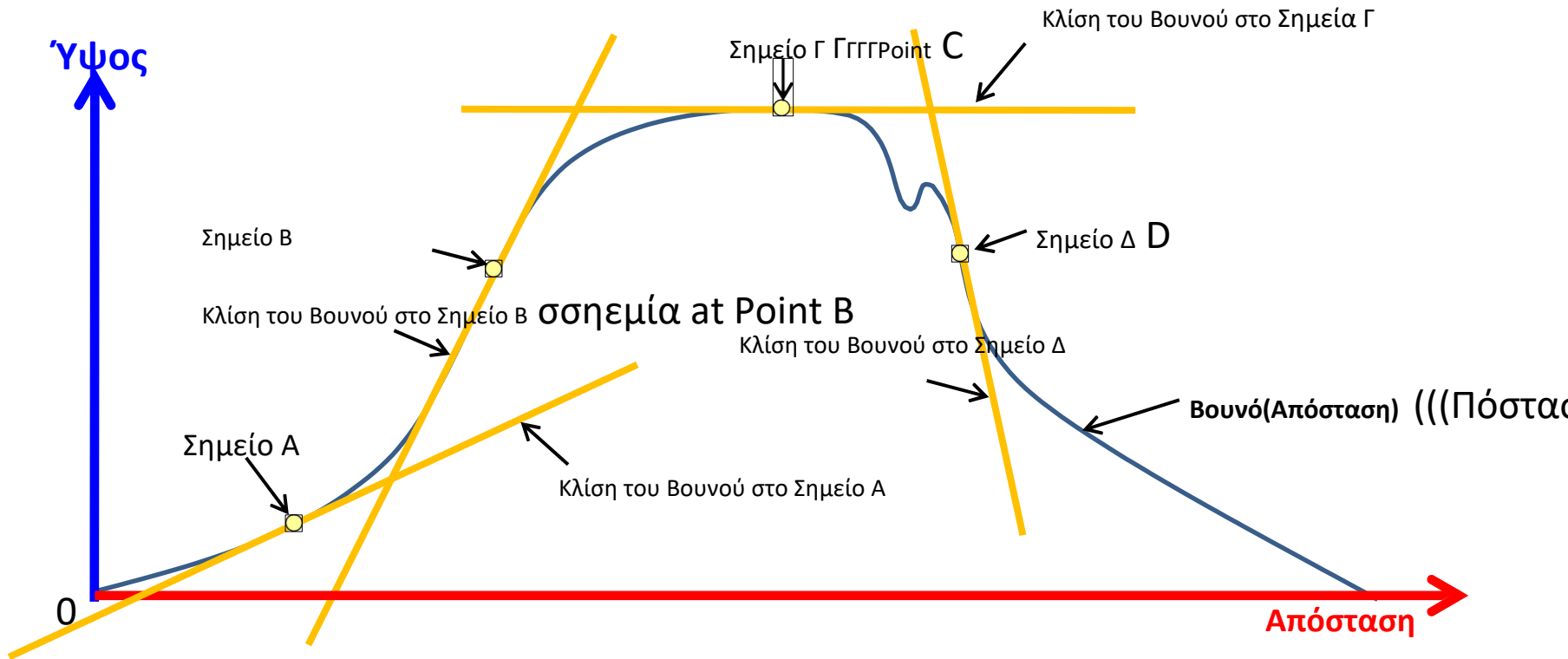
Γράφημα του βουνού του Jack και της Jill Hill με ύψος σε σχέση με την απόσταση.

Υπολογισμοί με τον Jack και την Jill



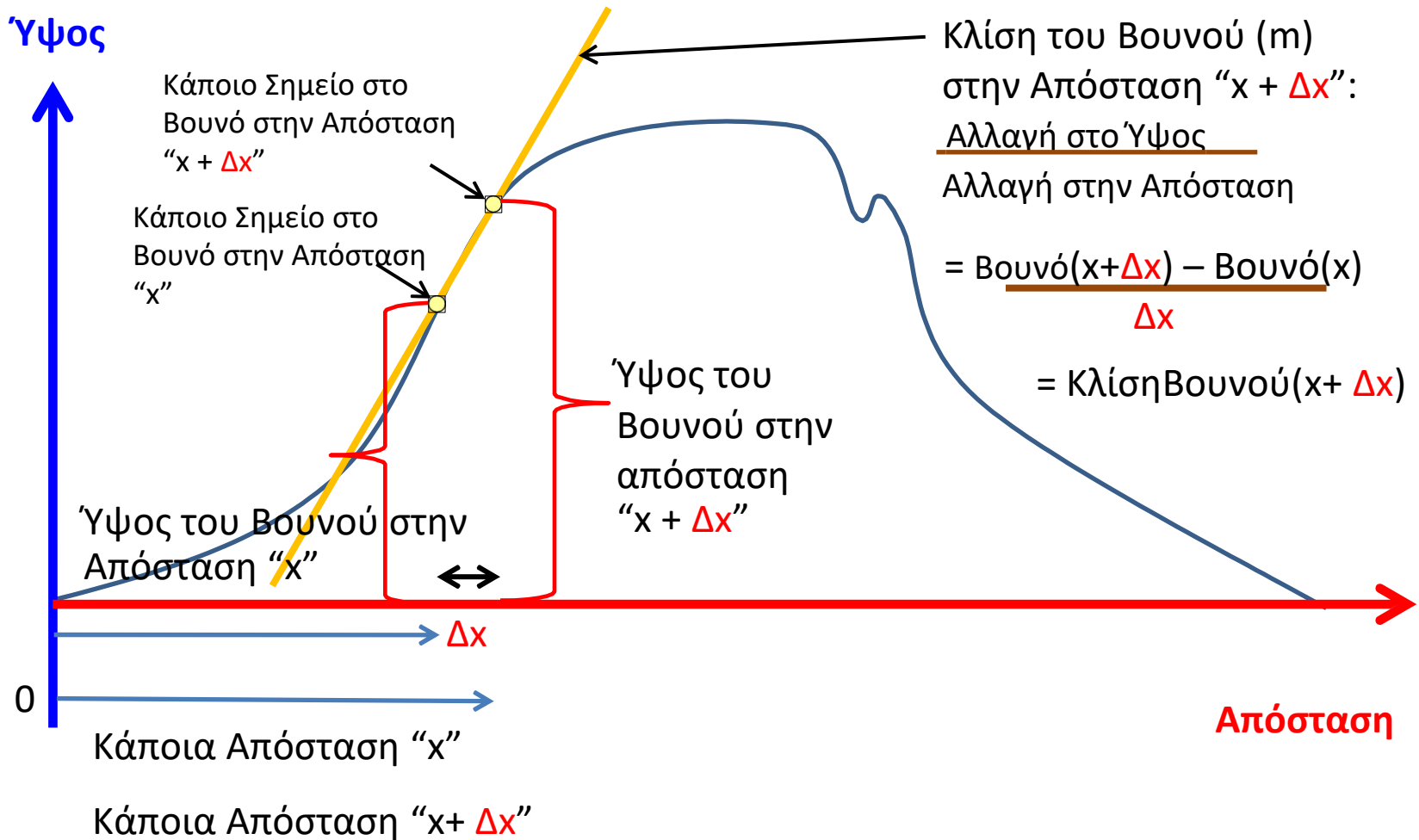
Απεικονίζουμε το Ύψος σε σχέση με την Απόσταση χρησιμοποιώντας μια Συνάρτηση.

Υπολογισμοί με τον Jack και την Jill



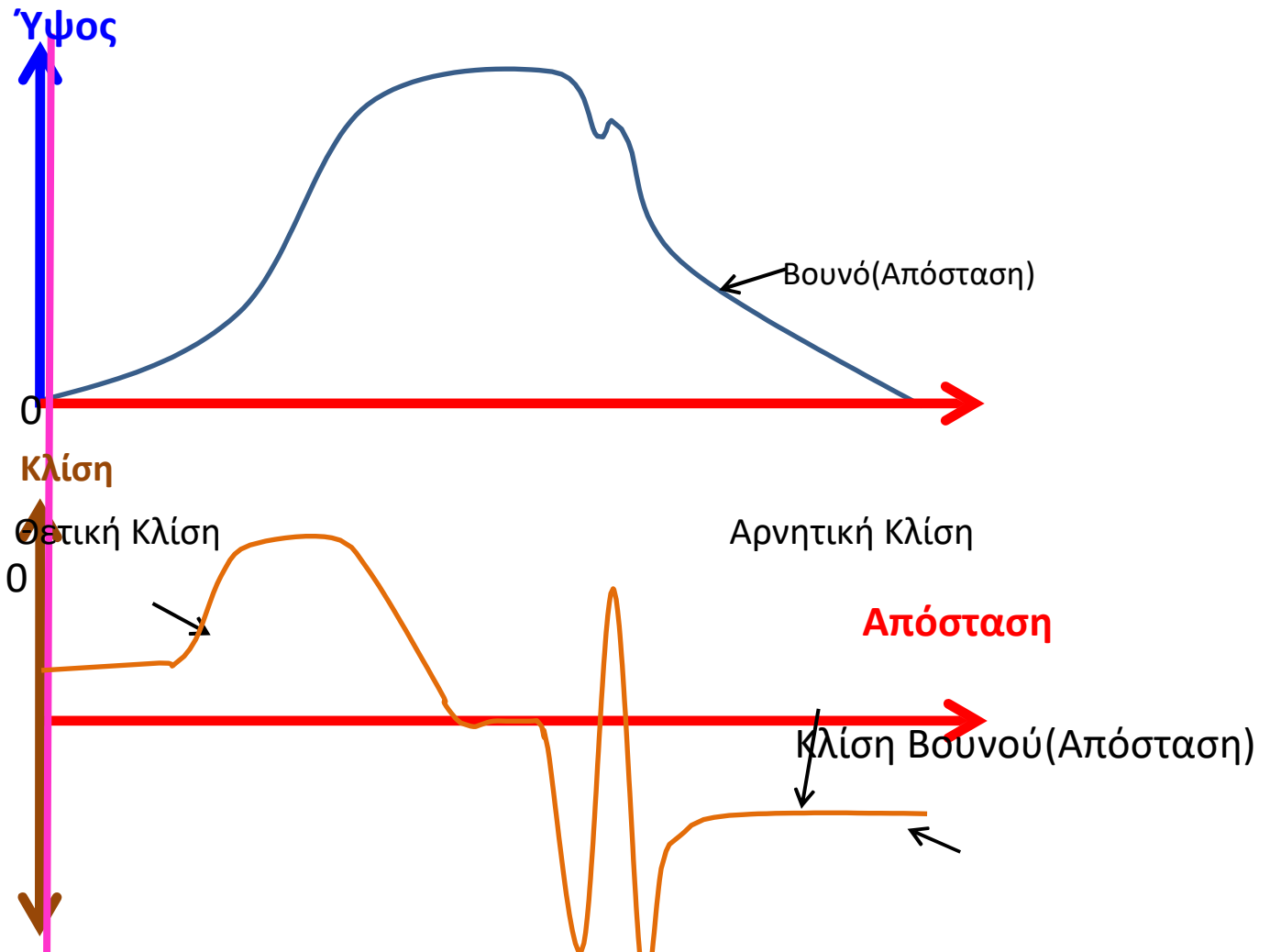
Το Βουνό έχει μία κλίση (m) σε κάθε σημείο της επιφάνειας του.

Υπολογισμοί με τον Jack και την Jill



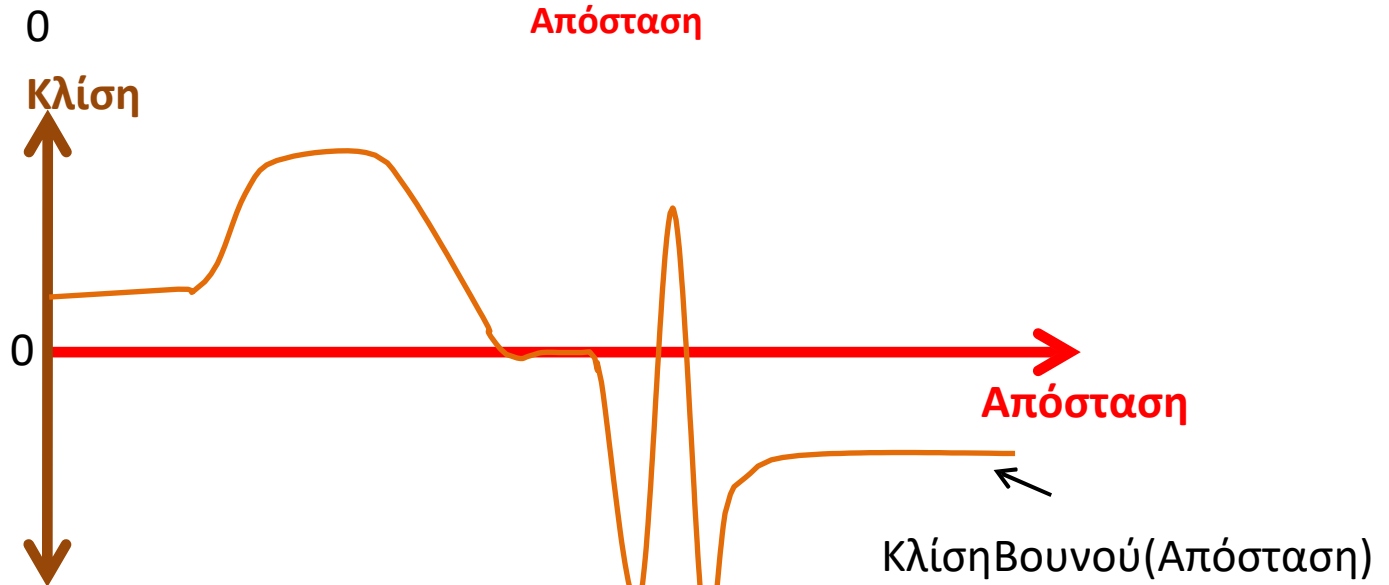
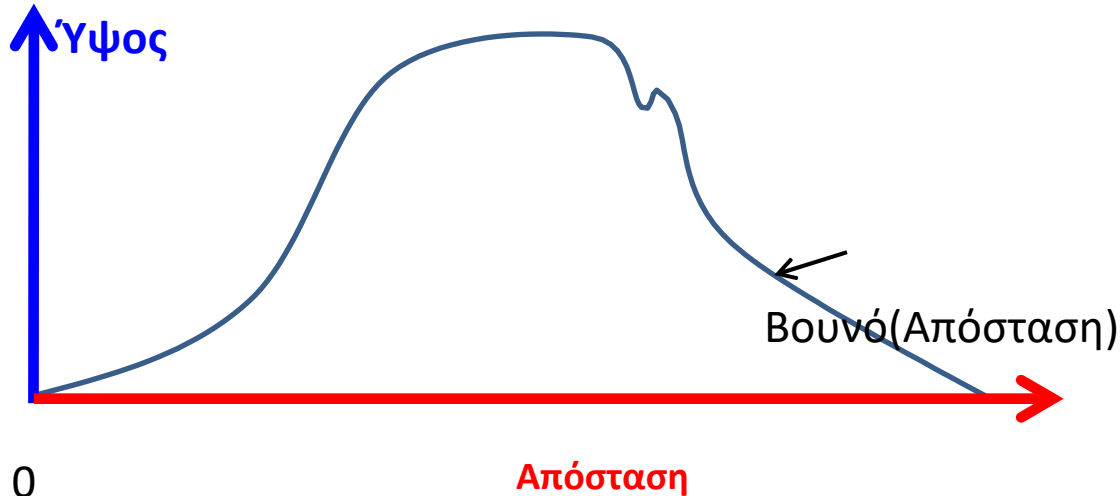
Μπορούμε προσεγγιστικά να υπολογίσουμε την κλίση χρησιμοποιώντας δύο σημεία της καμπύλης του Βουνού.

Υπολογισμοί με τον Jack και την Jill



Μπορούμε να σχεδιάσουμε την κλίση του λόφου σε ένα άλλο γράφημα!

Υπολογισμοί με τον Jack και την Jill

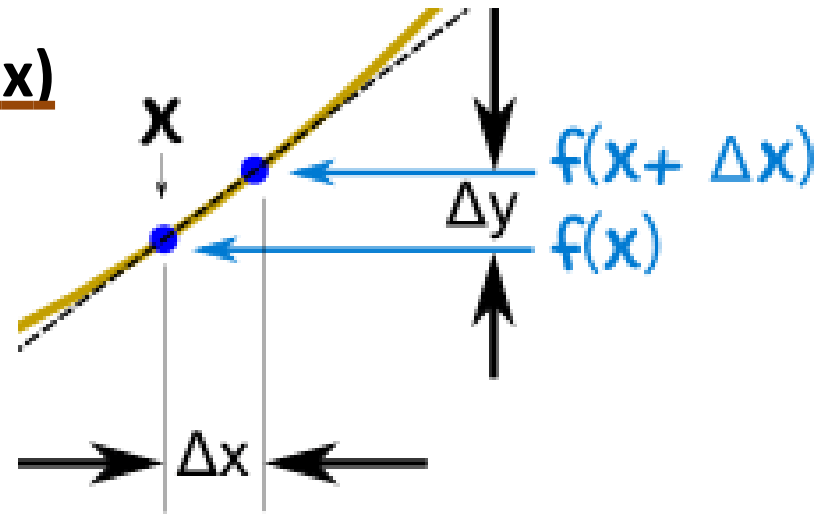


Η Κλίση Βουνού() είναι η Παράγωγος (Derivative) Συνάρτηση του Βουνού(!)

Υπολογισμοί με τον Jack και την Jill

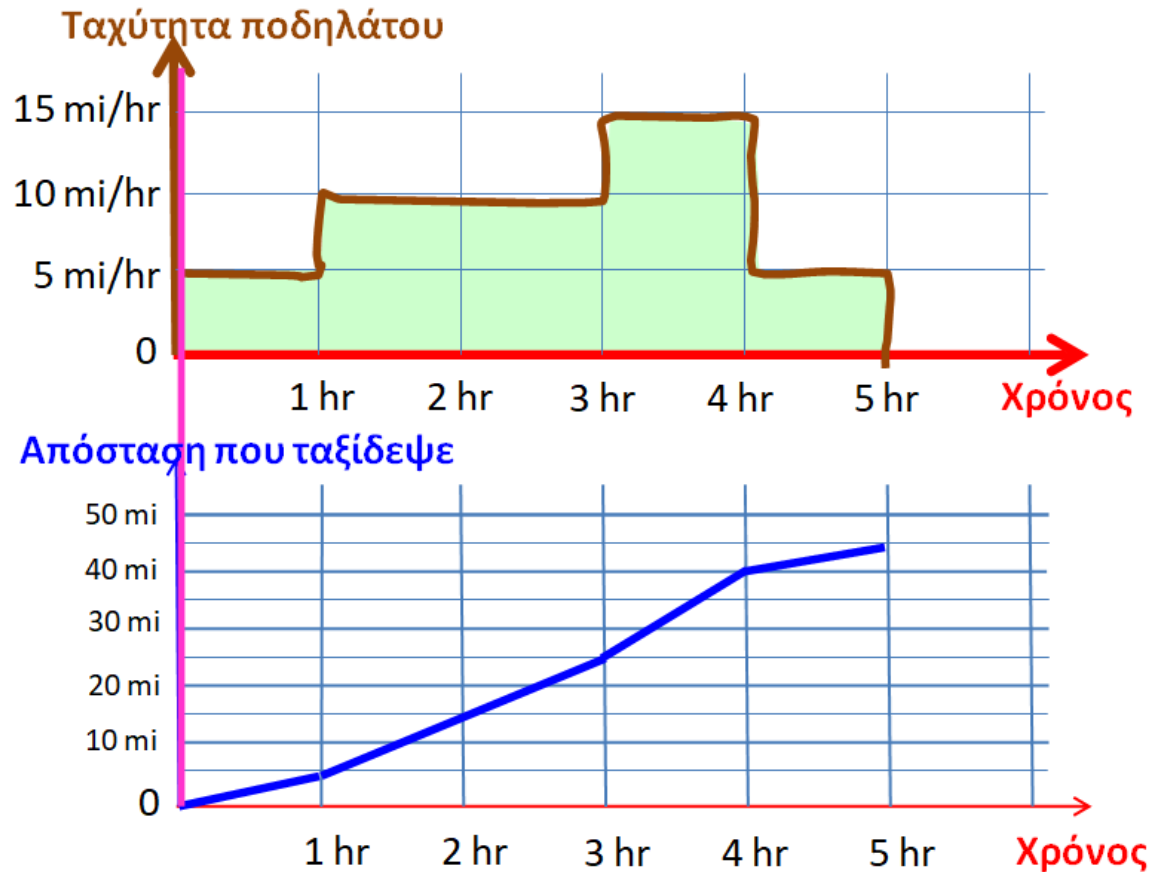
Έτσι, για να ανακεφαλαιώσουμε, η παράγωγος συνάρτηση οποιασδήποτε συνάρτησης, ας πούμε, $f(x)$, είναι απλώς η κλίση της συνάρτησης σε κάθε σημείο, την οποία μπορούμε να προσεγγίσουμε ως:

$$\text{Παράγωγος του } f(x) = \frac{f(x+\Delta x) - f(x)}{\Delta x}$$



Υπολογισμοί με τον Jack και την Jill

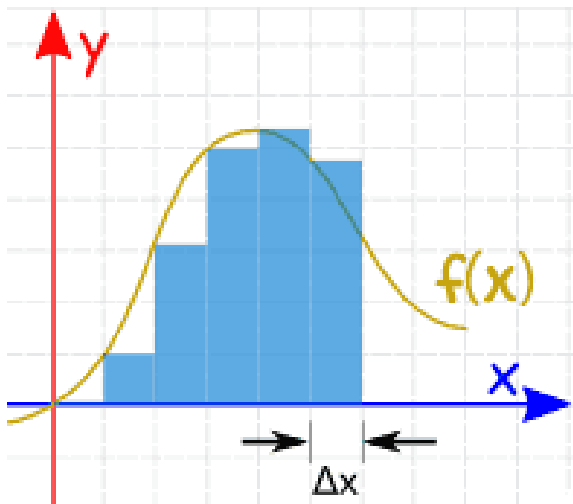
Το Ολοκλήρωμα (Integral) είναι το αντίστροφο της Παραγώγου (Derivative). Εάν έχουμε μια συνάρτηση παραγώγου, μπορούμε να την ολοκληρώσουμε για να βρούμε τη συνάρτηση αυτής του Ολοκληρώματος.



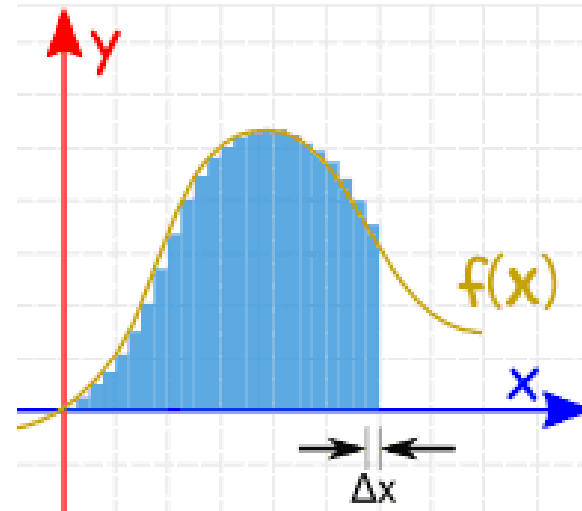
Υπολογισμοί με τον Jack και την Jill

Άρα, ολοκλήρωση είναι απλώς η εύρεση της περιοχής κάτω από οποιαδήποτε συνάρτηση, ας πούμε $f(x)$, την οποία μπορούμε να προσεγγίσουμε ως:

$$\text{Ολοκλήρωμα του } f(x) = \sum f(x) \cdot \Delta x$$

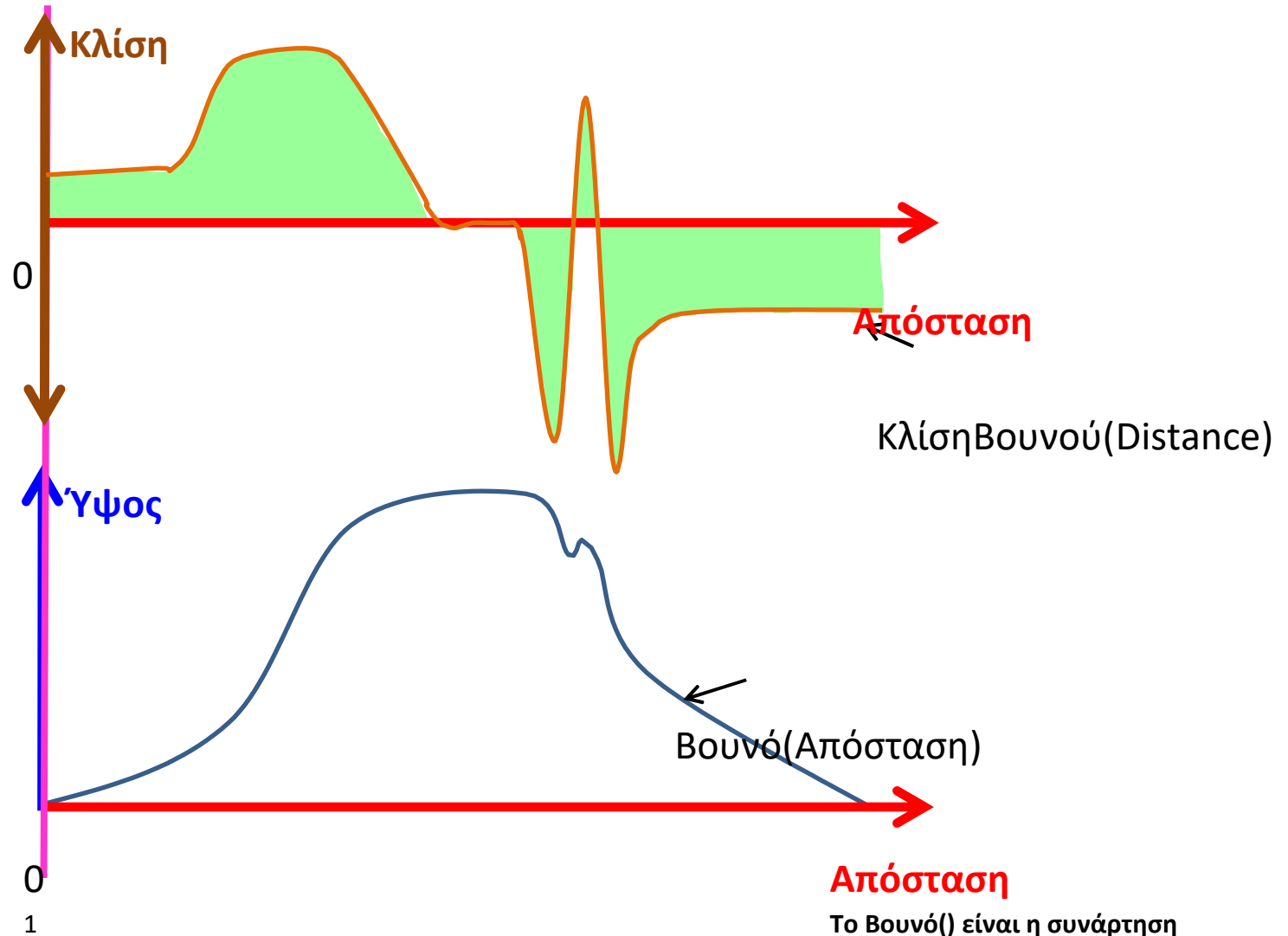


Προσθέτοντας «φέτες» μιας συνάρτησης $F(x)$ κάνοντας βήμα Δx .



Προσθέτοντας περισσότερες «φέτες» μιας συνάρτησης δίνει καλύτερα αποτελέσματα.

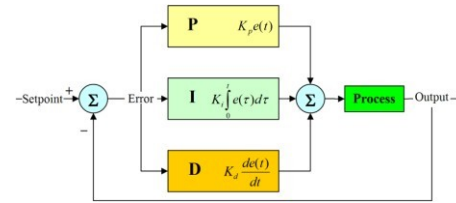
Υπολογισμοί με τον Jack και την Jill



Απόσταση

Το Βουνό() είναι η συνάρτηση Ολοκληρώματος του ΚλίσηΒουνού(!)

Τι είναι το PID?



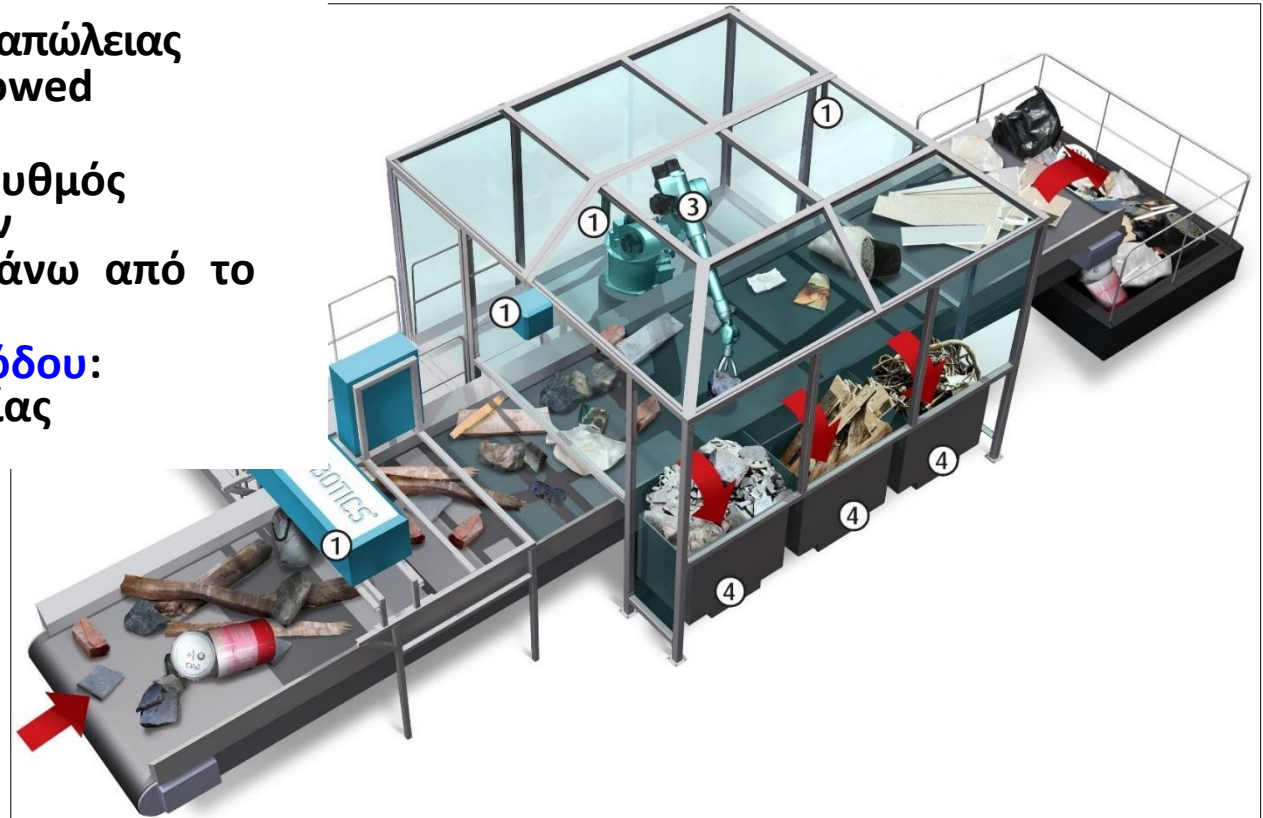
Το PID σημαίνει **Proportional (Αναλογικός)**, **Integral (Ολοκλήρωμα)**, και **Differential (Διαφορικός)** (P-I-D). Είναι η πιο κοινή μέθοδος ελέγχου κλειστού βρόχου που χρησιμοποιείται για τον έλεγχο πραγματικών πραγμάτων όπως η θερμοκρασία και η ταχύτητα του cruise control* ενός αυτοκινήτου. Για παράδειγμα, όταν το σπίτι σας είναι πολύ ζεστό, χαμηλώνετε τον θερμοστάτη και ένας ελεγκτής PID ενεργοποιεί και απενεργοποιεί το κλιματιστικό σας για να διατηρήσει το σπίτι σας στη νέα θερμοκρασία που έχετε ορίσει.

*Μια ηλεκτρονική συσκευή σε ένα μηχανοκίνητο όχημα που μπορεί να ενεργοποιηθεί για να διατηρήσει μια επιλεγμένη σταθερή ταχύτητα χωρίς τη χρήση γκαζιού.

Παραδείγματα του PID Control

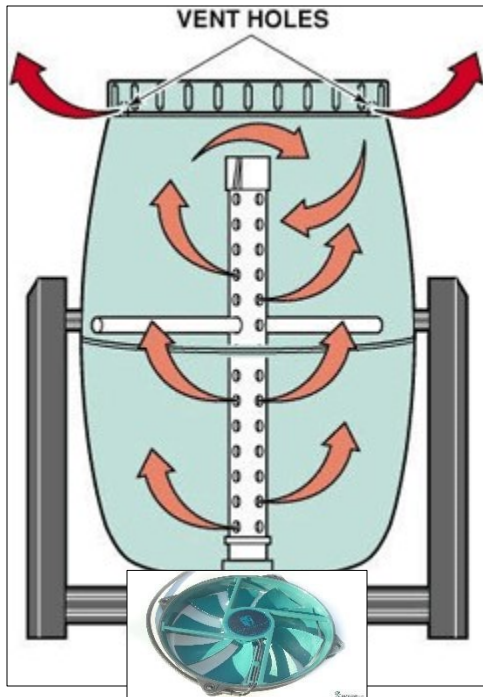
Ταχύτητα μεταφορικής ταινίας διαλογής απορριμμάτων

- **Έξοδος:** Απωλεσθέντα από την ταξινόμηση σκουπίδια
- **Σημείο Ρύθμισης (Set Point):** Ρυθμός επιτρεπόμενης απώλειας σκουπιδιών Allowed Missed
- **Λάθος (Error):** Ρυθμός απωλεσθέντων σκουπιδιών πάνω από το Set Point
- **Διαδικασία Εξόδου:** Ταχύτητα ταινίας



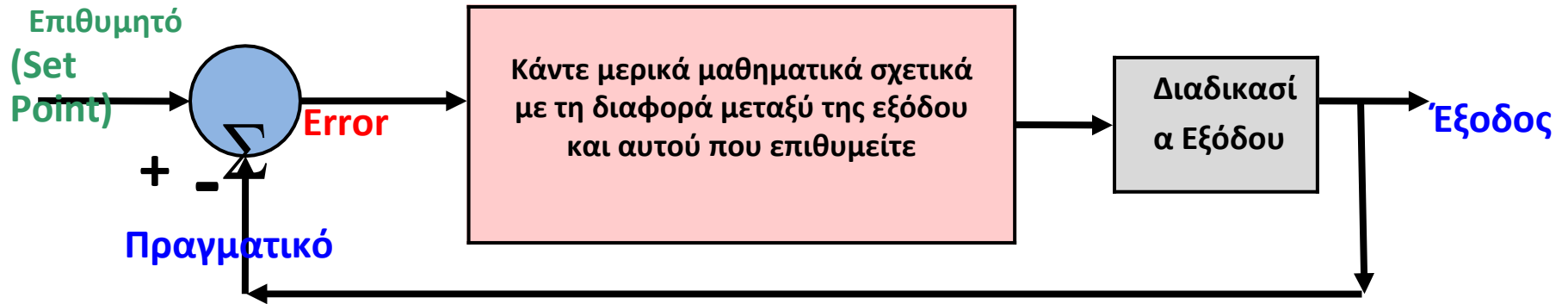
Ένα άλλο παράδειγμα του PID Control

Ένας ανεμιστήρα αερισμού για Κομπόστ



- **Έξοδος:** Θερμοκρασία του Κομπόστ
- **Set Point:** Η ιδανική θερμοκρασία του Κομπόστ
- **Error:** Η διαφορά στη θερμοκρασία του Κομπόστ
- **Διαδικασία εξόδου:** Η θερμοκρασία του ανεμιστήρα αερισμού

Ένας βρόχος ελέγχου ανάδρασης

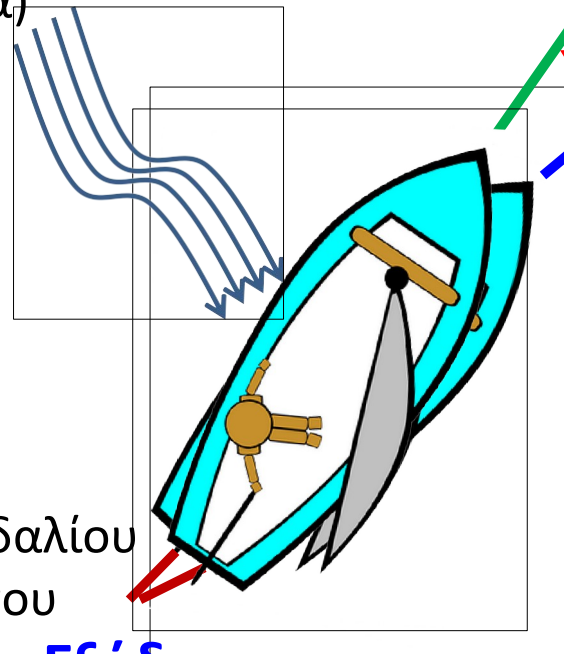


$\Delta T = 0$ χρόνος κάθε υπολογισμού του βρόχου

Ένας βρόχος ελέγχου ανάδρασης λαμβάνει την πραγματική έξοδο και τη συγκρίνει με την επιθυμητή έξοδο (βρίσκοντας το σφάλμα) με κάποιο ρυθμό βρόχου και προσπαθεί να κάνει την πραγματική έξοδο να ταιριάζει με την επιθυμητή.

Σκεφτείτε ένα ιστιοφόρο

Άνεμος (Μεταβάλλει την κατεύθυνση και την ταχύτητα)



Χειρισμός πηδαλίου του πλεούμενου
= **Διαδικασία Εξόδου**

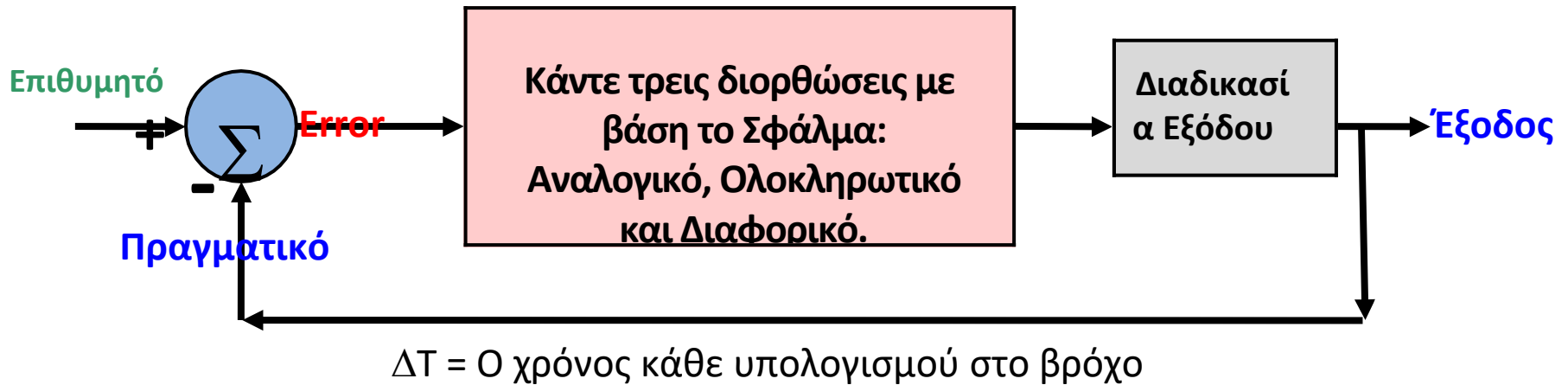
Επιθυμητή Κατεύθυνση Ταξιδιού
= **Set Point**

Πραγματική Κατεύθυνση Ταξιδιού
= **Έξοδος**

Error στην κατεύθυνση ταξιδιού

Σε ένα ιστιοφόρο, ο άνθρωπος που ελέγχει το πηδάλιο είναι η διαδικασία βρόχου ελέγχου ανάδρασης για να κατευθύνει το σκάφος στη σωστή κατεύθυνση. Καθώς ο άνεμος αλλάζει κατεύθυνση και ταχύτητα, ο άνθρωπος αντισταθμίζει αλλάζοντας τη θέση του πηδαλίου με βάση το σφάλμα στην κατεύθυνση που πηγαίνει το σκάφος.

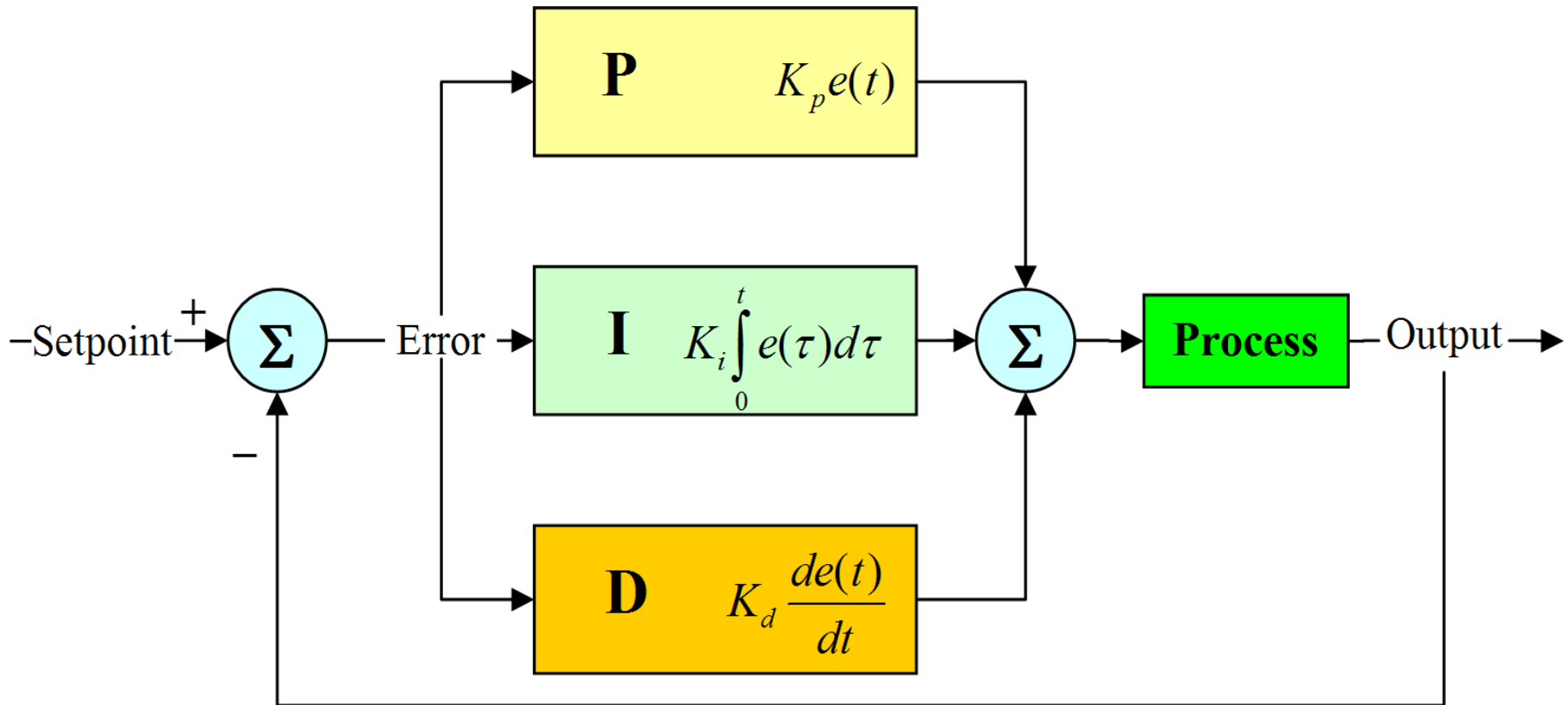
Βρόχος ελέγχου ανάδρασης PID



Ένας βρόχος ελέγχου ανάδρασης **PID** χρησιμοποιεί τον **Λάθος (Error)** όρο με τρεις διαφορετικούς τρόπους:

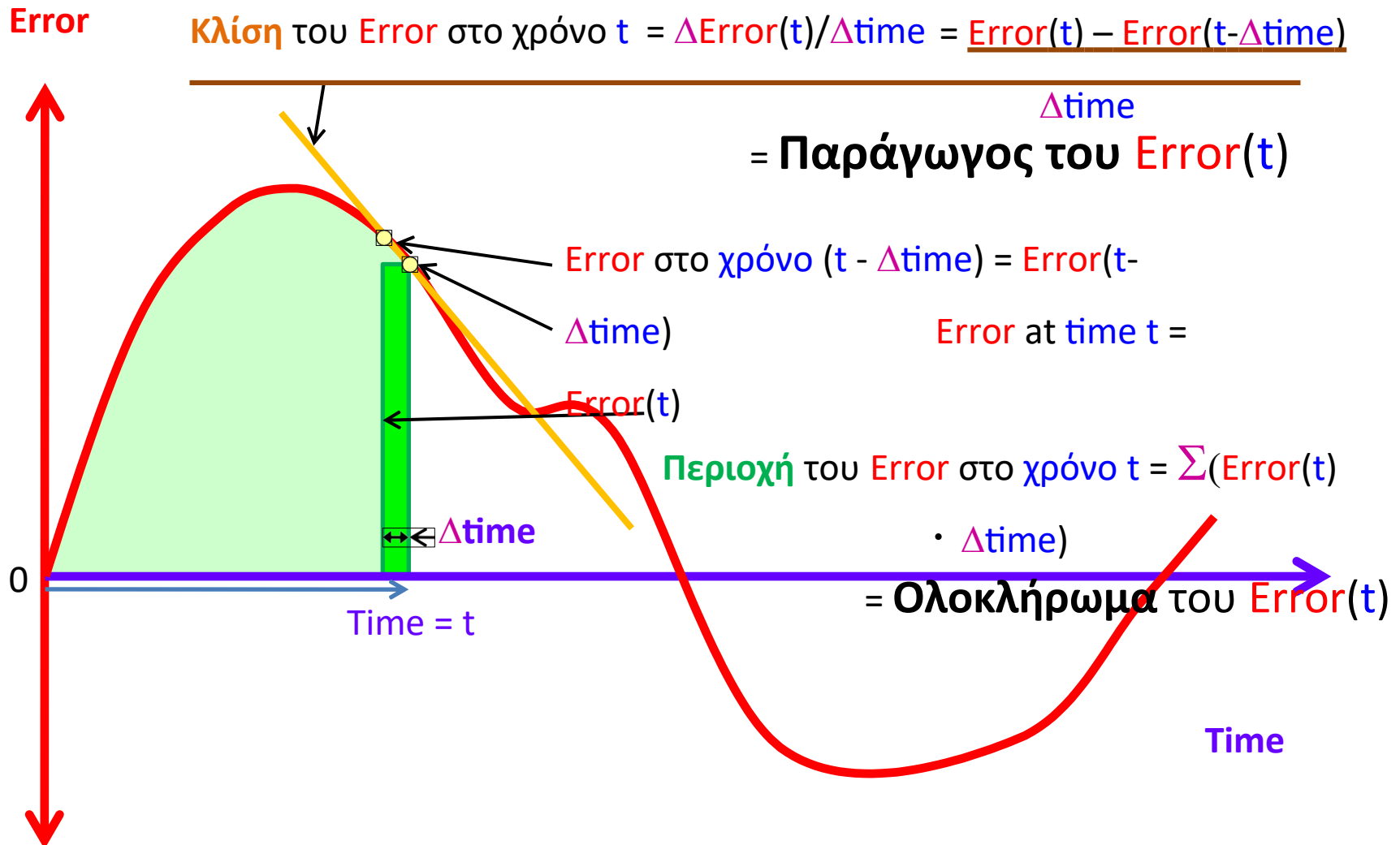
- Ο **Proportional (αναλογικός)** όρος αλλάζει την έξοδο αναλογικά με το Λάθος
- Ο **Integral (ολοκλήρωμα)** όρος προσθέτει όλα τα προηγούμενα λάθη (**error περιοχή**) και αντισταθμίζει την Έξοδο
- Ο **Differential (διαφορικός)** όρος επηρεάζει την έξοδο βασισμένος στο τρέχον λάθος (**κλίση** του **error**)

Το γενικό μπλοκ διάγραμμα PID



Αμάν! Τι είναι όλες αυτές οι ανοησίες; Υπολογισμοί!

Γραφική αναπαράσταση του Λάθους



Ο Proportional (Αναλογικός) όρος

Proportional

$$= K_p \cdot \text{Error}$$

K_p είναι ο αναλογικός συντελεστής (σταθερά) Καθορίζει πόσο αυτό το στοιχείο του ελεγκτή PID επηρεάζει την έξοδο.

Error είναι το **λάθος** στην **έξοδο** σε σχέση με το **επιθυμητό**. Υπολογίζεται ως η διαφορά του επιθυμητού set point και του τρέχοντος πραγματικού σημείου.

Ο Integral (Ολοκλήρωμα) όρος

$$\text{Integral} = K_i \cdot \Sigma (\text{Error} \cdot \Delta T)$$

K_i είναι ο συντελεστής του ολοκληρώματος (σταθερά) . Καθορίζει πόσο αυτό το στοιχείο του ελεγκτή PID επηρεάζει την έξοδο.

Σ είναι το ελληνικό σύμβολο που χρησιμοποιούμε για να δείξουμε άθροιση. Σε αυτήν την περίπτωση θέλουμε να αθροίσουμε (να προσθέσουμε) όλα τα σφάλματα από τότε που ξεκινήσαμε τον ελεγκτή PID.

Error είναι το **Λάθος** στην **έξοδο** σε σχέση με το **επιθυμητό**. Υπολογίζεται ως η διαφορά του επιθυμητού set point και του τρέχοντος πραγματικού σημείου.

ΔT είναι ο χρόνος από την τελευταία φορά που πραγματοποιήσαμε αυτόν τον υπολογισμό. Δ είναι το ελληνικό σύμβολο Δέλτα που χρησιμοποιούμε και σημαίνει αλλαγή. T είναι το γράμμα που χρησιμοποιούμε για να σημαίνει Χρόνος. Άρα ΔT είναι η αλλαγή στο χρόνο από τότε που κάναμε αυτόν τον υπολογισμό τελευταία. Είναι ο χρόνος δειγματοληψίας της τιμής σφάλματος.

Ο Διαφορικός (Differential) όρος

Differential

$$= K_d \cdot (\Delta \text{Error} / \Delta T)$$

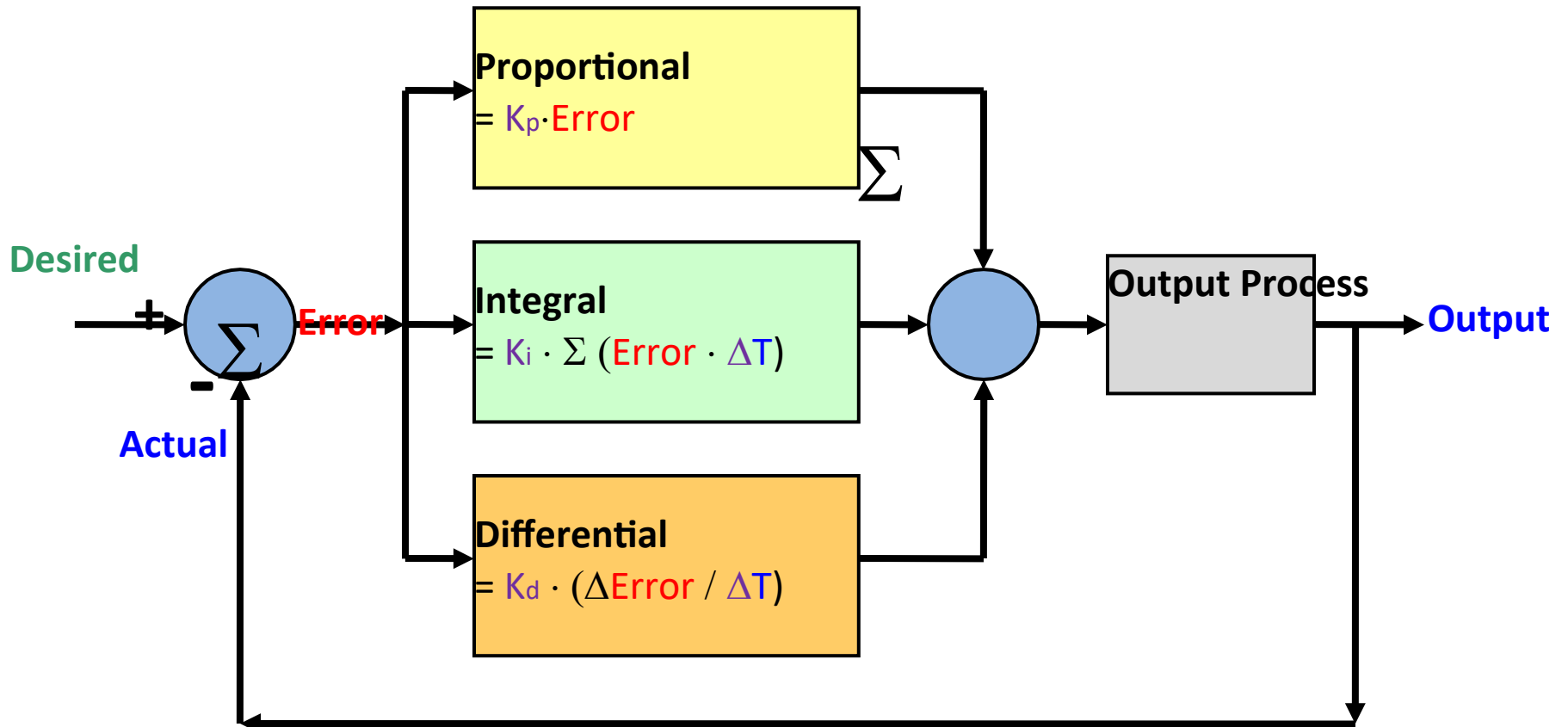
K_d είναι ο Διαφορικός συντελεστής (σταθερά). Καθορίζει πόσο αυτό το διαφορικό στοιχείο του ελεγκτή PID επηρεάζει την έξοδο.

Δ είναι ένα ελληνικό σύμβολο δέλτα που χρησιμοποιούμε για να δείξουμε αλλαγή. Εδώ, θέλουμε να υπολογίσουμε την αλλαγή από την τελευταία φορά που κάναμε αυτόν τον υπολογισμό.

ΔError είναι αλλαγή στο error της εξόδου από το επιθυμητό. Υπολογίζεται ως η διαφορά ανάμεσα στο τρέχον error and the error της τελευταίας φορές που κάναμε στον υπολογισμό.

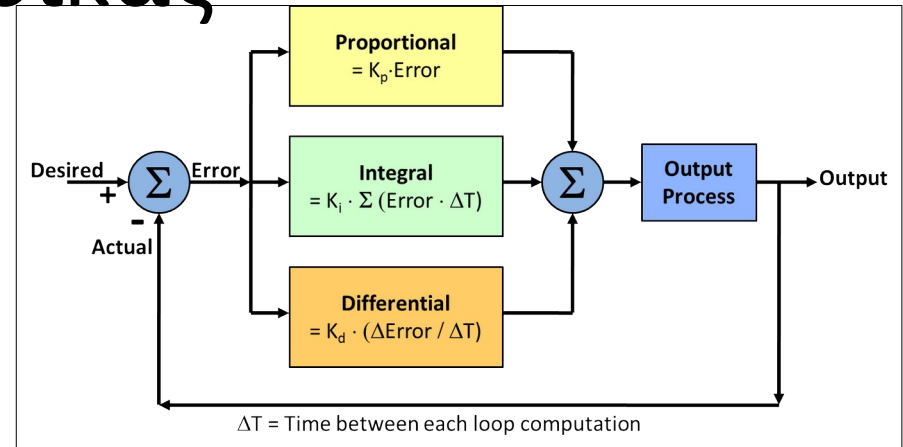
ΔT είναι ο χρόνος από την τελευταία φορά που πραγματοποιήσαμε αυτόν τον υπολογισμό. T είναι το γράμμα που χρησιμοποιούμε για να δείξουμε Χρόνο. Άρα ΔT είναι η αλλαγή στο χρόνο από τότε που κάναμε αυτόν τον υπολογισμό τελευταία. Είναι ο χρόνος δειγματοληψίας του σφάλματος.

Συναρμολογώντας το PID



ΔT = Ο χρόνος ανάμεσα σε κάθε υπολογισμό του βρόχου

PID Ψευδοκώδικας



```
Previous_Error = 0  
Integral       = 0  
DeltaT        = 1/100
```

Loop:

```
Error = Desired - Actual
```

```
Integral = Integral + (Error * DeltaT)
```

```
Derivative = (Error - Previous_Error) / DeltaT
```

```
Previous_Error = Error
```

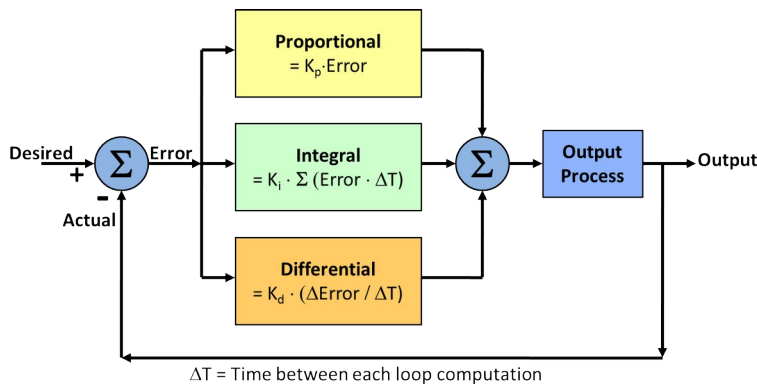
```
Output = ((Kp*Error) + (Ki*Integral) + (Kd*Derivative))
```

```
Wait(DeltaT)
```

Repeat Loop

Δεν μπορείς να συντονίσεις τα ψάρια...

Το να έχεις κωδικό PID στα χέρια σου είναι σαν να έχεις ένα ολοκαίνουριο εργοστασιακό, ακούρδιστο πιάνο



=



- Είναι τελείως άχρηστο ως μουσικό όργανο
- Μέχρι να κουρδιστεί!



Βρίσκοντας τις παραμέτρους K_p , K_i , and K_d

Μέθοδος Ziegler-Nichols

Το δύσκολο μέρος της χρήσης ενός ελεγκτή PID είναι η εύρεση των συντελεστών (K_p , K_i και K_d) για καθένα από τα τρία συνεισφέροντα στοιχεία για μια συγκεκριμένη χρήση ελεγκτή. Η εύρεση αυτών των συντελεστών απαιτεί επαναληπτικά (επαναλαμβανόμενα) πειράματα με τον ελεγκτή. Αυτό ονομάζεται «συντονισμός» (tuning). Μια κοινή μέθοδος συντονισμού ονομάζεται μέθοδος Ziegler-Nichols που πήρε το όνομά της από αυτούς που την επινόησαν

Ziegler-Nichols Method

Βήμα 1: Ορίστε τους συντελεστές K_i και K_d στο μηδέν. Έτσι, ο ελεγκτής γίνεται μόνο ελεγκτής τύπου P.

Βήμα 2: Ξεκινήστε με το K_p με "μικρή" τιμή και αυξήστε το K_p έως ότου η έξοδος του ελεγκτή αρχίσει να ταλαντώνεται (δεν καταλήγει σε τιμή εξόδου). Αυτή η τιμή κέρδους ονομάζεται κρίσιμο συντελεστής, K_c .

Βήμα 3: Μετρήστε τη χρονική περίοδο της ταλάντωσης όταν $K_p = K_c$. Αυτό ονομάζεται κρίσιμη περίοδος ταλάντωσης, P_c .

Βήμα 4: Χρησιμοποιήστε τις τιμές K_c και P_c για να προσδιορίσετε τα K_p , K_i και K_d χρησιμοποιώντας τους τύπους που προσδιόρισαν οι Ziegler και Nichols για έναν ελεγκτή P, PI ή PID (εμφανίζονται στην επόμενη διαφάνεια).

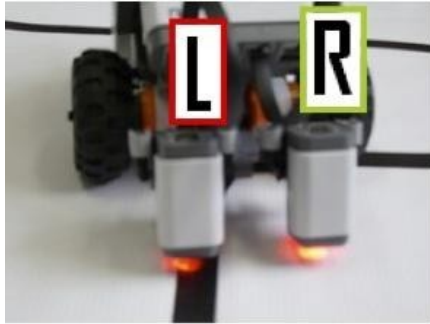
Μέθοδος των Ziegler-Nichols για τους συντελεστές

Control Type	K_p	K_i	K_d
P	$0.50K_c$	-	-
PI	$0.45K_c$	$1.2K_p / P_c$	-
PID	$0.60K_c$	$2.0K_p / P_c$	$K_p P_c / 8$

Αποτελέσματα της αύξησης των τιμών των παραμέτρων

Παράμετροι	Χρόνος ανόδου	Υπερχείλιση	Χρόνος τακτοποίησης	Σφάλμα στην ισορροπία
K_p	Μείωση	Αύξηση	Μικρή αλλαγή	Μείωση
K_i	Μείωση	Αύξηση	Αύξηση	Εξάλειψη
K_d	Αόριστο	Μείωση	Αύξηση	Τίποτα

Ακολουθητής γραμμής δύο αισθητήρων



Για μια γραμμή δύο αισθητήρων φωτός, εξετάστε:

- Ποιο είναι το επιθυμητό (Set Point);
- Ποιο είναι το Σφάλμα; Η διαφορά μεταξύ των τιμών φωτός φαίνεται; Τι γίνεται με τις διαφορές στους αισθητήρες;
- Πώς κατευθύνετε το ρομπότ; Χρησιμοποιήστε μπλοκ κινητήρα τιμονιού ή ξεχωριστά μπλοκ τροφοδοσίας κινητήρα;
- Ποιος είναι ο χρόνος δειγματοληψίας σας (ΔT , δηλ. χρόνος βρόχου);
- Ποιες είναι οι μεταβλητές σας; Ποιες είναι οι σταθερές σου;

PID 2 Light Sensor Line Follower Ψευδοκώδικας

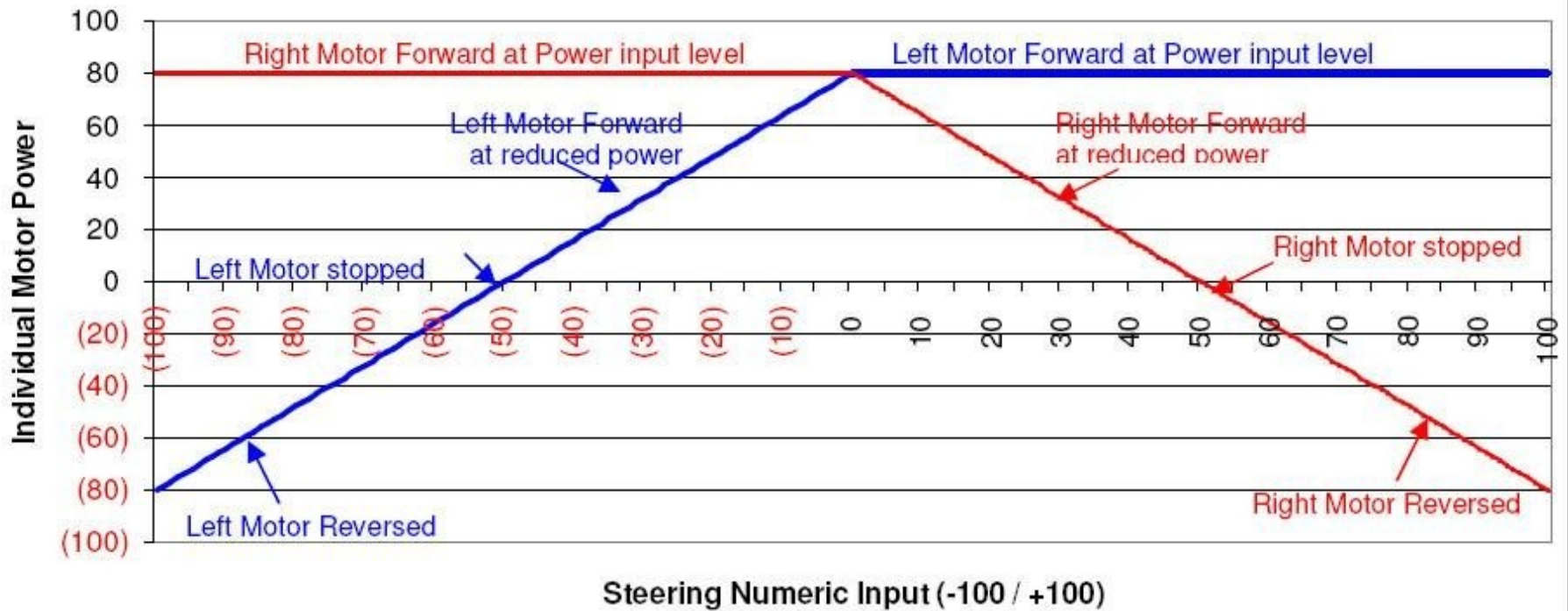
```
Previous_Error = 0
Integral= 0
DeltaT= 1/100 Light_Sensor_Correction = ???

Loop:
Get Left_Light_Value Get Right_Light_Value
Error = Left_Light_Value - Right_Light_Value - Light_Sensor_Correction Integral
Derivative = (Error - Previous_Error)/DeltaT Previous_Error = Error
Steering = (Kp*Error) + (Ki*Integral) + (Kd*Derivative) Wait(deltaT)
Repeat Loop
```

EV3 Move Steering Block Power

EV3 Forward Move Steering Block Motor Power Function

(Blue=left motor power, Red=right motor power)



— Left Motor Power, L — Right Motor Power, R

Αναφορές

- <http://www.mathisfun.com/>
- http://www.inpharmix.com/jps/PID_Controller_For_Lego_Mindstorms_Robots.html
- <http://thetechnicgear.com/2014/03/howto-create-line-following-robot-using-mindstorms/>
- https://en.wikipedia.org/wiki/PID_controller
- <http://controlguru.com/table-of-contents/>
- <http://www.wired.co.uk/magazine/archive/2013/06/start/a-smarter-way-to-sort-your-recycling>
- <http://www.urban-composting.com/>