

ΤΕΧΝΟΛΟΓΙΚΟ ΕΚΠΑΙΔΕΥΤΙΚΟ ΙΔΡΥΜΑ ΣΕΡΡΩΝ
ΣΧΟΛΗ ΤΕΧΝΟΛΟΓΙΚΩΝ ΕΦΑΡΜΟΓΩΝ
ΤΜΗΜΑ ΜΗΧΑΝΙΚΩΝ ΠΛΗΡΟΦΟΡΙΚΗΣ ΤΕ

Πτυχιακή εργασία

**Κατασκευή πακέτου προσομοίωσης σε Matlab της κυκλικής κίνησης
στην Φυσική**

Επιβλέπων Καθηγητής: Δρ. Απόστολος Κουιρουκίδης

Όνομα Σπουδαστή: Ιωάννης Χάγιαννης

Σέρρες 2015

ΠΕΡΙΕΧΟΜΕΝΑ

ΕΙΣΑΓΩΓΗ.....	2
1. ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗ	3
1.1. Η σημασία της προσομοίωσης.....	3
1.2. Η σημασία της οπτικοποίησης.....	3
2. ΚΥΚΛΙΚΗ ΚΙΝΗΣΗ	4
2.1. Νόμοι του Νεύτωνα	4
2.2. Ομαλή Κυκλική κίνηση	4
2.3. Κεντρομόλος δύναμη.....	5
3. ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB.....	6
3.1. Βασικές Πράξεις και Ορισμός Μεταβλητών	6
3.2. M-Files.....	7
3.3. Προγραμματισμός στο Matlab	8
3.4. Δημιουργία GUI στο Matlab.....	10
4. ΕΚΤΕΛΕΣΗ ΠΡΟΣΟΜΟΙΩΣΗ ΚΥΚΛΙΚΗΣ ΚΙΝΗΣΗΣ.....	12
4.1. Θεωρητικός σχεδιασμός εφαρμογής.....	12
4.2. Σχεδιασμός εφαρμογής	12
4.3. Προγραμματισμός εφαρμογής.....	14
4.4. Εκτέλεση εφαρμογής.....	15
ΠΑΡΑΡΤΗΜΑ.....	21
Α. Κώδικας MATLAB.....	21
ΒΙΒΛΙΟΓΡΑΦΙΑ	38

ΕΙΣΑΓΩΓΗ

Η κυκλική κίνηση είναι μια κίνηση που παρατηρείται πολύ συχνά, όπως στην κίνηση των τροχών των ποδηλάτων και των αυτοκινήτων, στην περιστροφή ενός ανεμιστήρα, στις κινήσεις πλανητών και δορυφόρων. Είναι γενικά εύκολο να περιγράψουμε και να μελετήσουμε τα καθαρά κινητικά χαρακτηριστικά της, όπως ακτίνα, περίοδο, συχνότητα, γωνιακή και γραμμική ταχύτητα και τις σχέσεις τους, αλλά μάλλον αδύνατο να παρατηρήσουμε την αιτία της, δηλαδή τα δυναμικά της χαρακτηριστικά, όπως την κεντρομόλο δύναμη και τις σχέσεις της με όλα τα προηγούμενα. Η μόνη εναλλακτική δυνατότητα που έχουμε, αν θέλουμε να κάνουμε την πειραματική μελέτη της δυναμικής της, είναι να καταφύγουμε σε προσομοιώσεις.

Στην εργασία αυτή θα μελετήσουμε την κυκλική κίνηση μίας σημειακής μάζας και ιδιαίτερα την εξάρτηση της κεντρομόλου δύναμης από τη μάζα, την ακτίνα της τροχιάς και τη γωνιακή ταχύτητα, δημιουργώντας ένα εικονικό εργαστήριο φυσικής μέσω του λογισμικού πακέτου Matlab.

1. ΠΡΟΣΟΜΟΙΩΣΗ ΚΑΙ ΟΠΤΙΚΟΠΟΙΗΣΗ

1.1. Η σημασία της προσομοίωσης

Η προσομοίωση (simulation) είναι η 'τεχνητή μίμηση' πραγματικών καταστάσεων, διαδικασιών ή πραγμάτων. Στις Φυσικές Επιστήμες, συνήθως αναπαριστούμε με την προσομοίωση κάποια ουσιώδη χαρακτηριστικά ή συμπεριφορές ενός φυσικού συστήματος. Η προσομοίωση την υλοποίηση μιας πειραματικής διαδικασίας, όχι όμως σε πραγματικό εργαστήριο, αλλά σε εικονικό. Σε ένα τέτοιο εικονικό εργαστήριο η 'λειτουργία' της φύσης γίνεται από έναν ηλεκτρονικό υπολογιστή.

Μερικά πλεονεκτήματα της προσομοίωσης είναι η επαναληψιμότητα όσον αφορά στα αποτελέσματα, η δυνατότητα εκτέλεσης του εικονικού πειράματος σε οποιοδήποτε περιβάλλον, η ευκολία χειρισμού μεταβλητών του περιβάλλοντος ή των αλληλεπιδράσεων, η ασφάλεια.

Τελευταίο στάδιο της προσομοίωσης, είναι η ανάλυση των αποτελεσμάτων, που συχνά γίνεται με την οπτικοποίησή τους.

1.2. Η σημασία της οπτικοποίησης

Δεν υπάρχει αμφιβολία ότι ο ανθρώπινος νους αντιλαμβάνεται πιο εύκολα και πιο γρήγορα το περιβάλλον του όταν το παρατηρεί σαν εικόνα, παρά όταν προσπαθεί να μεταφράσει τα αποτελέσματα της προσομοίωσης. Γίνεται έτσι απαραίτητο τα αποτελέσματα κάθε προσομοίωσης, που είναι αρχικά αριθμοί και που αντιστοιχούν σε συμβάντα, αντικείμενα και κατανομές στο χώρο, να «μεταφράζονται» σε εικόνες του συστήματος που μελετάμε και των παραμέτρων του. Με αυτές τις εικόνες οδηγούμαστε στην ταχύτερη και καλύτερη κατανόηση των φαινομένων. Στην επιστήμη, γενικότερα, χρησιμοποιούμε ένα πολύ μεγάλο πλήθος εικόνων και αναπαραστάσεων για την οπτικοποίηση.

2. ΚΥΚΛΙΚΗ ΚΙΝΗΣΗ

2.1. Νόμοι του Νεύτωνα

Οι δύο πρώτοι νόμοι του Νεύτωνα μας επιτρέπουν να περιγράψουμε την κίνηση που κάνει ένα σώμα όταν γνωρίζουμε τη συνισταμένη των δυνάμεων που ενεργούν σ' αυτό, την αρχική θέση του καθώς και την αρχική του ταχύτητα.

Έτσι αν σε ένα σώμα δεν ασκούνται δυνάμεις, ή αν ασκούνται και έχουν συνισταμένη μηδέν, τότε σύμφωνα με τον πρώτο νόμο του Νεύτωνα αυτό θα ηρεμεί ή θα κινείται με ομαλή ευθύγραμμη κίνηση.

Αν η συνισταμένη των δυνάμεων που ασκούνται σε ένα σώμα δεν είναι μηδέν, τότε σύμφωνα με το δεύτερο νόμο του Νεύτωνα αυτό έχει επιτάχυνση a , ομόρροπη της δύναμης, που προσδιορίζεται από τη σχέση $F = m \cdot a$, όπου m είναι η μάζα του σώματος.

2.2. Ομαλή Κυκλική κίνηση

Ένα κινητό κάνει κυκλική κίνηση όταν η τροχιά που διαγράφει είναι περιφέρεια κύκλου. Ομαλή χαρακτηρίζεται η κυκλική κίνηση ενός κινητού, όταν η τιμή της ταχύτητάς του παραμένει σταθερή.

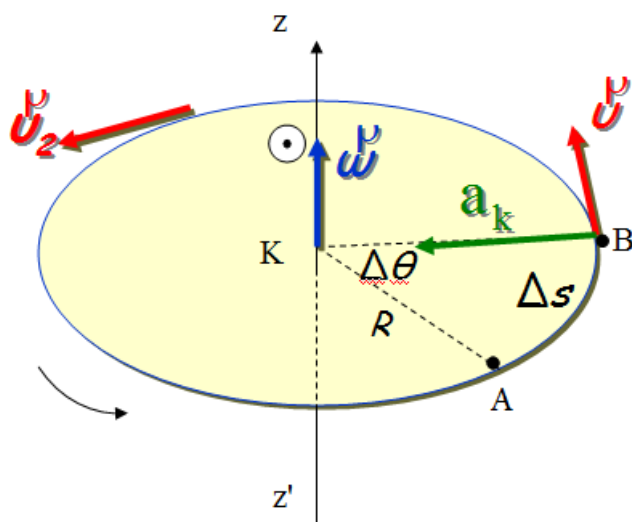
Ο χρόνος που χρειάζεται το κινητό για να κάνει μία περιφορά, λέγεται *περίοδος* (T) της κυκλικής κίνησης.

Ο αριθμός των περιφορών που εκτελεί το κινητό στη μονάδα του χρόνου λέγεται *συχνότητα* (f) της κυκλικής κίνησης.

Γραμμική ταχύτητα (u) είναι ένα διάνυσμα που έχει μέτρο ίσο με το πηλίκο του τόξου Δs που διανύει το υλικό σημείο σε χρόνο Δt προς τον χρόνο αυτό.

Γωνιακή ταχύτητα (ω) είναι ένα διάνυσμα που έχει διεύθυνση κάθετη στο επίπεδο της τροχιάς του και μέτρο ίσο με το πηλίκο της γωνίας $\Delta \theta$ που διαγράφει το υλικό σημείο σε χρόνο Δt προς τον χρόνο αυτό.

Η *κεντρομόλος επιτάχυνση* (a) είναι υπεύθυνη για τη μεταβολή της διεύθυνσης της γραμμικής ταχύτητας.



$$v = \frac{\Delta s}{\Delta t} = \frac{s}{t} = \frac{2\pi R}{T} = 2\pi Rf$$

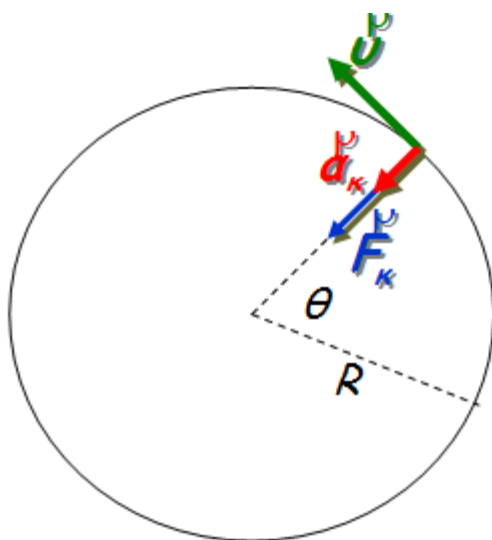
$$\omega = \frac{\Delta \theta}{\Delta t} = \frac{\theta}{t} = \frac{2\pi}{T} = 2\pi f$$

$$a_k = \frac{v^2}{R} = \omega^2 R$$

2.3. Κεντρομόλος δύναμη

Ας θεωρήσουμε την περίπτωση που ένα σώμα εκτελεί κυκλική κίνηση με ταχύτητα σταθερής τιμής. Επειδή η κατεύθυνση της ταχύτητας συνεχώς μεταβάλλεται, συνεπώς υπάρχει επιτάχυνση (κεντρομόλος) και σύμφωνα με το δεύτερο νόμο του Νεύτωνα στο σώμα ασκείται δύναμη. Η δύναμη αυτή έχει κατεύθυνση προς το κέντρο της κυκλικής τροχιάς και γι' αυτό λέγεται κεντρομόλος δύναμη

Γενικά κάθε δύναμη που αναγκάζει ένα σώμα να εκτελεί ομαλή κυκλική κίνηση λέγεται κεντρομόλος δύναμη. Η κεντρομόλος επιτάχυνση έχει την ίδια κατεύθυνση με την κεντρομόλο δύναμη.



$$F_k = m \cdot a_k = m \cdot \frac{v^2}{R}$$

3. ΕΙΣΑΓΩΓΗ ΣΤΟ MATLAB

Το λογισμικό MATLAB, παίρνει το όνομά του από τις λέξεις MATrix LABoratory, είναι ένα σύγχρονο ολοκληρωμένο μαθηματικό πακέτο για υψηλής απόδοσης αριθμητικούς υπολογισμούς (numerical computations). Είναι ένα διαδραστικό (interactive) πρόγραμμα για αριθμητικούς υπολογισμούς και για κατασκευή γραφημάτων, αλλά παρέχει επίσης και τη δυνατότητα προγραμματισμού, κάτι που το καθιστά ένα χρήσιμο εργαλείο για όλους όσους ασχολούνται με τις θετικές επιστήμες.

Αποτελεί ένα εξελιγμένο υπολογιστικό εργαλείο το οποίο μπορεί να βρει εφαρμογή σε διάφορους τομείς της επιστήμης αλλά βέβαια και της πράξης, όπως για παράδειγμα τη μηχανική, την ιατρική και τη βιομηχανική παραγωγή. Μάλιστα, το φάσμα των εφαρμογών του συγκεκριμένου πακέτου λογισμικού διευρύνεται συνεχώς και περισσότερο, αναδεικνύοντας με αυτό τον τρόπο τις πολλαπλές δυνατότητες του, όπως την υψηλή απόδοση και ταχύτητα υπολογιστικών αναλύσεων, την δυνατότητα προσομοίωσης φυσικών συστημάτων, την δυνατότητα υλοποίησης αλγορίθμων, την δυνατότητα αμφίδρομης επικοινωνίας με πληθώρα άλλων προγραμμάτων και εφαρμογών και την υψηλής ποιότητας γραφικές απεικονίσεις και animations.

3.1. Βασικές Πράξεις και Ορισμός Μεταβλητών

Η πιο απλή λειτουργία του Matlab είναι η χρήση του ως μία κοινή αριθμομηχανή. Η σειρά εκτέλεσης των πράξεων, στο Matlab είναι η ακόλουθη:

- Στην περίπτωση που σε μια αριθμητική παράσταση υπάρχουν παρενθέσεις εκτελούνται πρώτα οι πράξεις μέσα στις παρενθέσεις από μέσα προς τα έξω.
- Η εκτέλεση των πράξεων γίνεται πάντοτε από τα αριστερά προς τα δεξιά.
- Μεταξύ των βασικών αριθμητικών πράξεων πρώτα εκτελείται η ύψωση σε δύναμη.
- Στη συνέχεια εκτελούνται οι πολλαπλασιασμοί και οι διαιρέσεις με ίδια προτεραιότητα.
- Τελευταίες εκτελούνται οι προσθέσεις και οι αφαιρέσεις με την ίδια προτεραιότητα.

Μία εντολή εισάγεται πληκτρολογώντας στο παράθυρο εντολών δεξιά από το σύμβολο προτροπής ». Η εισαγωγή της εντολής ολοκληρώνεται με το πάτημα του πλήκτρου *Enter*. Στην περίπτωση που δεν ορίζεται κάποια μεταβλητή, η εντολή εκτελείται και το αποτέλεσμα εμφανίζεται στην οθόνη μετά την έκφραση *ans=* (συντομογραφία της αγγλικής λέξης *answer*). Η έκφραση *ans* αποτελεί μια προεπιλεγμένη μεταβλητή η οποία χρησιμοποιείται για να αποθηκευτεί το αποτέλεσμα της εντολής.

Στον ορισμό των μεταβλητών θα πρέπει να αναφέρουμε ότι μία μεταβλητή μπορεί να οριστεί χρησιμοποιώντας αλφαριθμητικούς χαρακτήρες και χαρακτήρες υπογράμμισης (_). Το Matlab δεν δέχεται ελληνικούς χαρακτήρες για τον ορισμό των μεταβλητών. Ο πρώτος χαρακτήρας μιας μεταβλητής θα πρέπει να είναι γράμμα του λατινικού αλφαβήτου. Ο μέγιστος αριθμός χαρακτήρων που μπορεί να έχει μία μεταβλητή είναι 63 χαρακτήρες.

Μία εντολή στο Matlab εκτός από αριθμούς, μεταβλητές και αριθμητικές παραστάσεις, μπορεί να περιλαμβάνει και συναρτήσεις. Το Matlab διαθέτει μία πληθώρα έτοιμων συναρτήσεων για την υλοποίηση ενός μεγάλου αριθμού μαθηματικών υπολογισμών. Κάθε συνάρτηση έχει ένα όνομα και μία παράμετρο (αριθμό, μεταβλητή, αριθμητική παράσταση) μέσα σε παρενθέσεις.

Το Matlab δίνει τη δυνατότητα στο χρήστη να εισάγει σχόλια μαζί με τις εντολές. Αυτό επιτυγχάνεται πληκτρολογώντας το σύμβολο *επί τοις εκατό* (%) στην αρχή μιας γραμμής ή δίπλα από μια εντολή. Οτιδήποτε γράφεται μετά το σύμβολο % αναγνωρίζεται από το Matlab ως σχόλιο.

3.2.M-Files

Το Matlab παρέχει τη δυνατότητα αποθήκευσης μιας ακολουθίας εντολών σε ένα αρχείο και στη συνέχεια την εκτέλεση αυτών με μία μόνο εντολή. Η ακολουθία αυτή των εντολών μπορεί να γραφεί σε ένα αρχείο κειμένου με την επέκταση (*m*), δηλαδή *όνομαΑρχείου.m*.

Στο Matlab υπάρχουν δύο διαφορετικοί τύποι M-Files. τα *αρχεία script* και τα *αρχεία συναρτήσεως*.

Τα αρχεία *script* :

- είναι ιδιαίτερα χρήσιμα σε περιπτώσεις που απαιτείται η επανάληψη μιας

μεγάλης ακολουθίας εντολών,

- δεν έχουν μεταβλητές εισόδου και εξόδου,
- οι μεταβλητές που υπολογίζονται προστίθενται στο χώρο εργασίας με αποτέλεσμα να μπορούν να χρησιμοποιηθούν και από άλλα αρχεία script ή από το παράθυρο εντολών. Πρόκειται δηλαδή για καθολικές μεταβλητές (global variables).

Τα αρχεία συναρτήσεων :

- επιτρέπουν τον εμπλουτισμό της γλώσσας προγραμματισμού Matlab στα πλαίσια των εφαρμογών του χρήστη,
- δέχονται μεταβλητές εισόδου και επιστρέφουν μεταβλητές εξόδου,
- οι μεταβλητές που ορίζονται μέσα στη συνάρτηση ισχύουν μόνο για τη συνάρτηση, είναι δηλαδή τοπικές μεταβλητές (local variables).

3.3. Προγραμματισμός στο Matlab

Το Matlab ως γλώσσα προγραμματισμού διαθέτει ένα πλούσιο σύνολο εργαλείων που μπορούν να χρησιμοποιηθούν για τον έλεγχο της ροής ενός προγράμματος. Ένα πρόγραμμα είναι μια ακολουθία εντολών που στην απλούστερη του μορφή οι εντολές εκτελούνται η μία μετά την άλλη, με τη σειρά που έχουν πληκτρολογηθεί. Τις περισσότερες φορές σε πιο σύνθετα προγράμματα, διαφορετικές μεταβλητές εισόδου απαιτούν την υπό συνθήκη εκτέλεση κάποιων τμημάτων εντολών του προγράμματος ή μπορεί να απαιτείται η επανάληψη μιας ακολουθίας εντολών.

Συνεπώς θα πρέπει να δηλώνεται κατά την πληκτρολόγηση του κώδικα ενός προγράμματος η πιθανή μεταβολή της ροής του προγράμματος. Αυτό πραγματοποιείται με τη χρήση των σχεσιακών και λογικών τελεστών καθώς και με τις δομές ελέγχου.

Η δομή if — end χρησιμοποιείτε στην περίπτωση που υπάρχει μια αμοιβαία αποκλειόμενη περίπτωση που καθορίζετε από την ισχύ μιας συνθήκης.

Η δομή if — else — end χρησιμοποιείτε στην περίπτωση που υπάρχουν δύο αμοιβαία αποκλειόμενες περιπτώσεις που καθορίζονται από την ισχύ μιας συνθήκης.

Η δομή if — else if — else — end χρησιμοποιείτε στην περίπτωση που υπάρχουν περισσότερες από δύο αμοιβαία αποκλειόμενες περιπτώσεις

Οι πιο συχνά χρησιμοποιούμενες δομές επανάληψης είναι ο βρόχος (loop) του *for* και ο βρόχος του *while*. Σε ένα βρόχο μία εντολή ή μια ομάδα εντολών επαναλαμβάνεται συνεχώς για έναν προκαθορισμένο αριθμό επαναλήψεων ή έως ότου ισχύσει μια συνθήκη. Σε κάθε επανάληψη, η οποία ονομάζεται βήμα, τουλάχιστον μία μεταβλητή η οποία ορίζεται μέσα στον βρόχο λαμβάνει νέα τιμή.

Ειδικότερα, ο βρόχος *for* χρησιμοποιείται για την εκτέλεση μιας εντολής ή μιας ομάδας εντολών για συγκεκριμένο αριθμό επαναλήψεων.

Από την άλλη, ο βρόχος *while* χρησιμοποιείται για την εκτέλεση μιας εντολής ή μιας ομάδας εντολών όταν δεν είναι προκαθορισμένο το πλήθος των επαναλήψεων αλλά εξαρτάται από την ισχύ μιας συγκεκριμένης συνθήκης

Η δομή *switch* επιτρέπει την επιλογή εκτέλεσης διαφορετικών συνόλων εντολών ανάλογα με την τιμή μιας έκφρασης. Ουσιαστικά είναι εναλλακτική της δομής *if –else if - else -end* για την περίπτωση όπου οι διάφορες ελεγχόμενες συνθήκες είναι συνθήκες ισότητας. Η δομή *switch* χρησιμοποιείται συνήθως για τη δημιουργία ενός μενού επιλογών.

Μερικές έτοιμες συναρτήσεις που είναι υλοποιημένες στη Matlab, τις οποίες χρησιμοποίησα για την υλοποίηση του προγράμματος είναι:

Hold on

Διατηρεί το τρέχον γράφημα και προσθέτει άλλο ένα γράφημα σε αυτό. Η MATLAB προσαρμόζει τα όρια στους άξονες και σημειώνει τις ετικέτες όπως είναι απαραίτητο για να εμφανιστεί το πλήρες φάσμα του γραφήματος που προστίθεται.

Hold off

Επαναφέρει την κατάσταση «hold» στην προεπιλεγμένη συμπεριφορά, στην οποία η MATLAB καθαρίζει το υπάρχον γράφημα και επαναφέρει τις ιδιότητες των αξόνων στις προεπιλογές τους πριν την σχεδίαση νέων γραφημάτων.

Str2double

Μετατρέπει τη συμβολοσειρά που δέχεται ως είσοδο σε μια αναπαράσταση πραγματικού αριθμού. Η συμβολοσειρά θα πρέπει να είναι σε μια αναπαράσταση χαρακτήρα τύπου ASCII που αντιστοιχεί σε ένα πραγματικό αριθμό.

3.4. Δημιουργία GUI στο Matlab

Το MATLAB προσφέρει στο χρήστη τη δυνατότητα να κατασκευάσει δικές του γραφικές διεπιφάνειες, (GUI).

Ένα GUI αποτελείται συνήθως από διάφορα παράθυρα, τα οποία περιέχουν ποικίλα στοιχεία ελέγχου όπως πεδία κειμένου, γραμμές κύλισης, λίστες, πεδία εισαγωγής κειμένου κ.α. Τα παράθυρα αυτά είναι δυνατόν να καλούν το ένα το άλλο, να δέχονται δεδομένα από το χρήστη, να μεταβιβάζουν πιθανώς τα δεδομένα από το ένα παράθυρο στο άλλο και γενικά να επιτελούν διάφορες λειτουργίες.

Η ανάπτυξη μίας εφαρμογής σε GUI χωρίζεται σε τέσσερα στάδια:

- Θεωρητική σχεδίαση της εφαρμογής πριν την υλοποίησή της σε MATLAB.
- Σχεδίαση της εφαρμογής στο κατάλληλο περιβάλλον (GUIDE Layout Editor) και δημιουργία των απαιτούμενων αντικειμένων.
- Καθορισμός των ιδιοτήτων του GUI και του κάθε αντικειμένου.
- Προγραμματισμός του κάθε αντικειμένου, εάν χρειάζεται.

Το GUIDE είναι το πρόγραμμα σχεδιασμού GUI της Matlab. Εμφανίζεται με την εντολή GUIDE και μας δίνει τη δυνατότητα να σχεδιάσουμε το GUI και να προσθέσουμε τον κώδικα.

Το GUIDE για να το πετύχει αυτό δημιουργεί για κάθε νέο παράθυρο δύο αρχεία. Τα αρχεία αυτά είναι το Fig-file και το M-file.

Το Fig-file ουσιαστικά είναι το παράθυρο–figure, όπου το MATLAB αποθηκεύει τα στοιχεία ελέγχου και την ακριβή θέση τους. Εδώ ο προγραμματιστής σχεδιάζει την εμφάνιση του παραθύρου.

Στο M-file ο προγραμματιστής πρέπει να γράψει τον κώδικα που θα ενσωματωθεί στα στοιχεία ελέγχου (π.χ. κουμπιά ή φόρμες εισαγωγής δεδομένων), ώστε αυτά να επιτελέσουν τις επιθυμητές λειτουργίες.

Πρέπει να σημειωθεί ότι κάθε αρχείο *.fig πρέπει να συνοδεύεται από το αντίστοιχο αρχείο *.m (με το ίδιο όνομα). Αν για κάποιο λόγο, χαθεί ή καταστραφεί το *.m αρχείο, τότε το παράθυρο *.fig δε θα είναι λειτουργικό. Κάθε φορά που κάποιος χρήστης δημιουργεί ένα νέο παράθυρο (figure), το

GUIDE δημιουργεί αυτομάτως και τους δυο προαναφερθέντες τύπους αρχείων.

4. ΕΚΤΕΛΕΣΗ ΠΡΟΣΟΜΟΙΩΣΗ ΚΥΚΛΙΚΗΣ ΚΙΝΗΣΗΣ

4.1. Θεωρητικός σχεδιασμός εφαρμογής

Η εφαρμογή που θα δημιουργήσουμε αφορά την μελέτη της κυκλική κίνησης μίας σημειακής μάζας, οπτικοποιώντας την εκτέλεση της κίνησης σε ρυθμό προσομοίωσης που επιλέγει ο χρήστης.

Οι πράξεις που θα υπολογίζει η εφαρμογή θα αφορούν την εξάρτηση της κεντρομόλου δύναμης από τη μάζα, την ακτίνα της τροχιάς και τη γωνιακή ταχύτητα.

Για την καλύτερη λειτουργία του συγκεκριμένου προγράμματος, οι τιμές στις μεταβλητές είναι εντός καθορισμένων οριακών τιμών.

Ο χρήστης θα χειρίζεται το πρόγραμμα μέσω μιας διεπιφάνειας, η οποία θα περιέχει τα απαραίτητα στοιχεία ελέγχου:

- 5 κουμπιά εκτέλεσης εντολών (`push_button`) για να εκτελούν τις λειτουργίες Εκτέλεση, Διακοπή, Ανανέωση, Έξοδος, Βοήθεια.
- 6 στοιχεία ελέγχου (`edit_text`). για πάρουμε την τιμή που θα εισάγει μέσα σε αυτό ο χρήστης.
- Δημιουργία γραφήματος (`axes`) σε πραγματικό χρόνο για οπτικοποίηση των αποτελεσμάτων.

4.2. Σχεδιασμός εφαρμογής

Αρχικά εκκινούμε το GUIDE και δημιουργούμε ένα νέο παράθυρο-figure. Το αποθηκεύουμε δίνοντάς του ταυτόχρονα το όνομα *gkentromolos*. Σε αυτό το σημείο το MATLAB αυτομάτως εκκινεί τον M-file editor και δημιουργεί το αντίστοιχο M-file. Παρατηρούμε ότι στο M-file δημιουργούνται κάποιες ενσωματωμένες συναρτήσεις, όπως είναι οι ακόλουθες:

```
function varargout = gkentromolos (varargin)
```

```
function gkentromolos _OpeningFcn(hObject, eventdata, handles, varargin)
```

```
function varargout = gkentromolos _OutputFcn(hObject, eventdata, handles)
```

Οι παραπάνω συναρτήσεις που ενσωματώθηκαν στο M-file, είναι απαραίτητες για τη σωστή λειτουργία του παραθύρου. Το MATLAB για κάθε μία από αυτές συμπεριλαμβάνει κάποια επεξηγηματικά σχόλια.

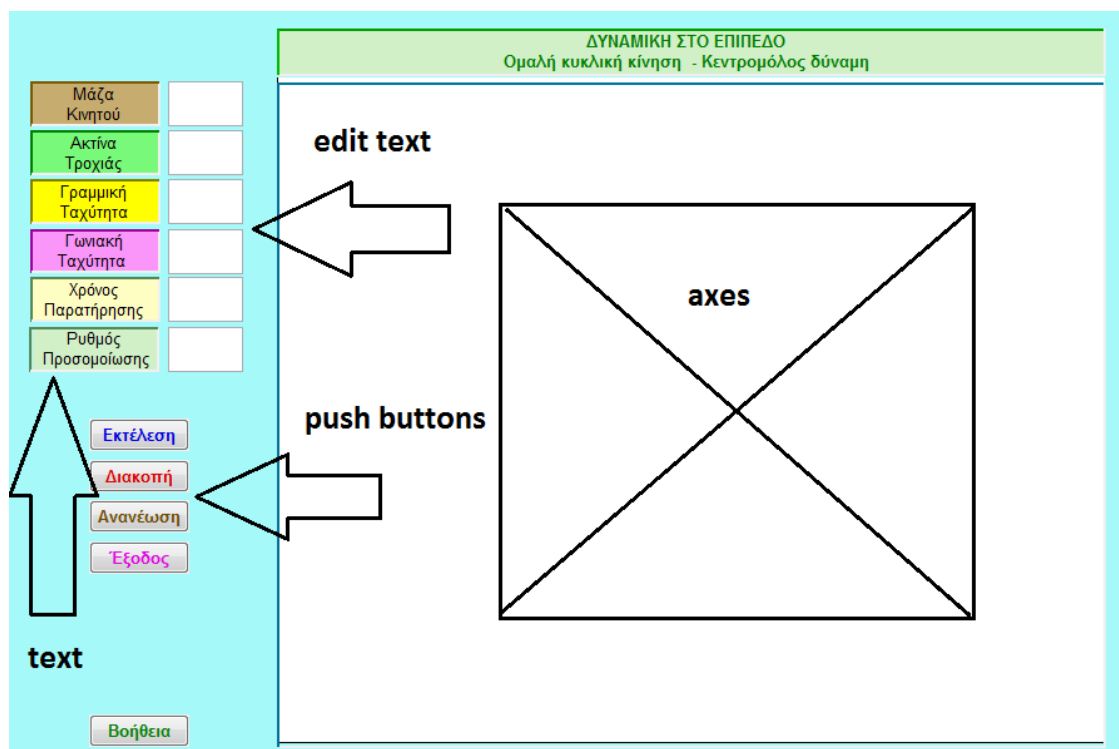
Συγκεκριμένα, η πρώτη συνάρτηση χρειάζεται ώστε να μπορεί ο χρήστης να καλεί το παράθυρο από τη γραμμή εντολών πληκτρολογώντας το όνομά του (> gkentromolos).

Επιπλέον, ο κώδικας που περιλαμβάνεται στο σώμα εντολών της δεύτερης συνάρτησης εκτελείται ακριβώς πριν εκκινήσει το παράθυρο.

Τέλος, οι εντολές που υπάρχουν στην τρίτη συνάρτηση, εξάγουν τα αποτελέσματά τους στη γραμμή εντολών του MATLAB.

Έπειτα θα αρχίσουμε να εισάγουμε από την αριστερή κάθετη γραμμή εργαλείων τα διάφορα στοιχεία ελέγχου. Εισάγουμε πρώτα ένα πλαίσιο (frame), αλλάζοντας το μέγεθός του έτσι ώστε να καλύπτει τον χώρο στον οποίο θα βάλουμε τα στοιχεία έλεγχο της εφαρμογής μας, καθώς και την διαμόρφωση του.

Στη συνέχεια εισάγουμε όλα στοιχεία ελέγχου που θα μας χρειαστούν. Αναλυτικά, πέντε push buttons, έξι edit text, έξι static text και axes.



Συνεχίζουμε αλλάζοντας, εάν θέλουμε, για όλα αυτά τα στοιχεία που έχουμε εισάγει το χρώμα τους κατά τη βούληση μας, καθώς επίσης και το κείμενο που θα φαίνεται επάνω τους αλλά και το μέγεθος των στοιχείων ελέγχου. Επιπλέον, όπως προαναφέραμε, μπορούμε να τα στοιχίσουμε για να πετύχουμε ομοιόμορφη εμφάνιση.

4.3. Προγραμματισμός εφαρμογής

Σε αυτό το σημείο στο αντίστοιχο M-file, παρατηρεί κανείς ότι έχουν ενσωματωθεί πολλές ακόμη συναρτήσεις. Στην συνέχεια θα αρχίσουμε τον προγραμματισμό των push buttons, τα οποία θα εκτελούν κάποιες λειτουργίες.

Μέσα στο M-file υπάρχουν οι αντίστοιχες συναρτήσεις :

function pushbutton1_Callback(hObject, eventdata, handles) %Εκτέλεση

function pushbutton2_Callback(hObject, eventdata, handles) % Διακοπή

function pushbutton3_Callback(hObject, eventdata, handles) %Ανανέωση

function pushbutton4_Callback(hObject, eventdata, handles) %Έξοδος

function pushbutton5_Callback(hObject, eventdata, handles) % Βοήθεια

Στην πρώτη θα δηλώσουμε τις εντολές που χρειάζονται για να εκτελεστεί η λειτουργία *Εκτέλεση*. Δηλαδή την εκτέλεση της εφαρμογής και την ενεργοποίηση των λειτουργιών *Ανανέωση* και *Έξοδος*.

Στην δεύτερη θα δηλώσουμε τις εντολές που χρειάζονται για να εκτελεστεί η λειτουργία *Διακοπή*. Δηλαδή την παύση εκτέλεσης της εφαρμογής και την ενεργοποίηση των λειτουργιών *Συνεχία*, *Ανανέωση* και *Έξοδος*.

Στην τρίτη θα δηλώσουμε τις εντολές που χρειάζονται για να εκτελεστεί η λειτουργία *Ανανέωση*. Δηλαδή να διαγράφεται το όποιο αποτέλεσμα της εφαρμογής και να εμφανίζονται κενά (Nul) τα edit texts, για να εισάγει ξανά ο χρήστης τις τιμές και την ενεργοποίηση την λειτουργίας *Εκτέλεση*.

Στην τέταρτη θα δηλώσουμε τις εντολές που χρειάζονται για να εκτελεστεί η λειτουργία *Έξοδος*, όπου θα κλείνει η εφαρμογή.

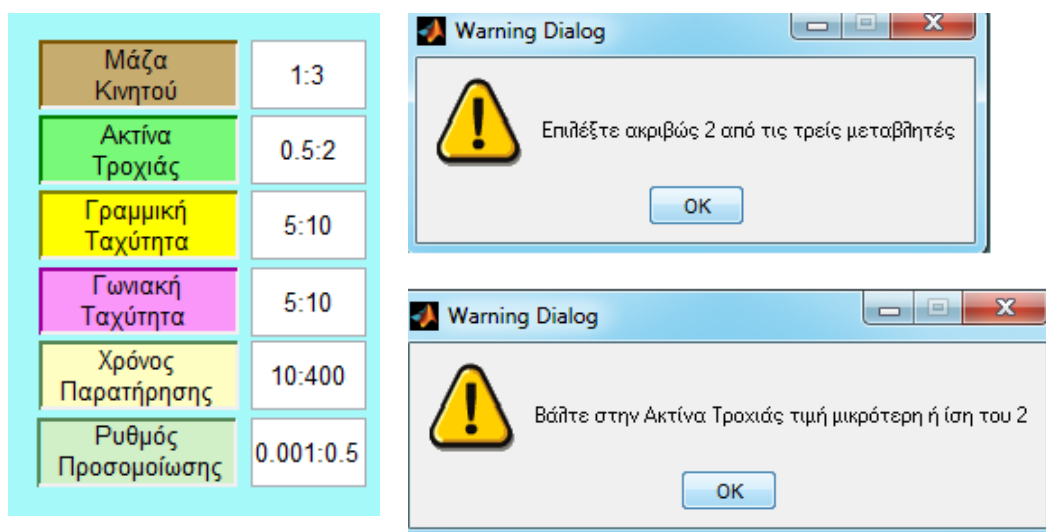
Στην πέμπτη θα δηλώσουμε τις εντολές που χρειάζονται για να εκτελεστεί η λειτουργία *Βοήθεια*, όπου θα παραπέμπει σε βοηθητικό αρχείο με επεξηγήσεις.

4.4. Εκτέλεση εφαρμογής

Κατά την εκτέλεση του προγράμματος εμφανίζεται το figure της εικονικής απεικόνισης της Κυκλικής Κίνησης με προκαθορισμένες τιμές στις μεταβλητές / παραμέτρους *Μάζα Κινητού (m)*, *Χρόνος Παρατήρησης (t)* και *Ρυθμός Προσομοίωσης*. Στις τιμές αυτές υπάρχει η δυνατότητα αλλαγής τους.



Στις μεταβλητές *Ακτίνα Τροχιάς (R)*, *Γραμμική Ταχύτητα(u)*, *Γωνιακή Ταχύτητα (ω)*, θα πρέπει να δώσουμε τιμές στις δύο από τις τρεις μεταβλητές στα όρια που έχουμε καθορίσει.



Στην περίπτωση που ο χρήστης δεν δηλώσει τις δυο από τις τρεις μεταβλητές ή καταχωρήσει τιμές σε οποιαδήποτε μεταβλητή εκτός των πιο πάνω ορίων, τότε εμφανίζεται μήνυμα λάθους, μέσω της εντολής :

`h=warndlg(warningstring).`

`if (R>2)`

`hfin=warndlg('Βάλτε στην Ακτίνα Τροχιάς τιμή μικρότερη ή ίση του 2');`

`return`

`end`

Η εντολή εμφανίζει ένα παράθυρο διαλόγου (dialog box) με το μήνυμα προειδοποίησης που είναι δηλωμένο στο string.

Η εντολή `h=questdlg(warningstring)` εμφανίζει παράθυρο διαλόγου (dialog box) με το μήνυμα προειδοποίησης. Το παράθυρο διαλόγου έχει τρία κουπιά, YES, NO, CANCEL και με την επιλογή τους δίνουν τον ανάλογο string στην h.

Ο χρήστης δηλώνει τιμές στις μεταβλητές που επιθυμεί, μέσω των edit text, και πατώντας το push button *Εκτέλεση*, η εφαρμογή αρχίζει να εκτελείται.

ΔΥΝΑΜΙΚΗ ΣΤΟ ΕΠΙΠΕΔΟ
Ομαλή κυκλική κίνηση - Κεντρομόλος δύναμη

Μάζα Κινητού	3
Ακτίνα Τροχιάς	1
Γραμμική Ταχύτητα	5
Γωνιακή Ταχύτητα	
Χρόνος Παρατήρησης	10
Ρυθμός Προσομοίωσης	0.5

Εκτέλεση
Συνέχεια
Ανανέωση
Έξοδος
Βοήθεια

$m = 3$	Kgm	$t = 10$	sec	$F_{\kappa} = \frac{mu^2}{R}$	(1)
$R = 1$	m			$u = \omega R$	(2)
$u = 5$	m/sec			$F_{\kappa} = m\omega^2 R$	(3)
$\omega = 5$	rad/sec			$F_{\kappa} = m\omega u$	(4)
$F_{\kappa} = 75$	Kgm/sec ²				

Η εφαρμογή εκτελείτε για χρόνο παρατήρησης (t) που έχει δηλώσει ο χρήστης. Κατά την διάρκεια της εκτέλεσης ο χρήστης έχει την δυνατότητα :

- να λάβει τις τιμές των φυσικών παραμέτρων της κυκλικής κίνησης καθώς και την τιμή της κεντρομόλου δύναμης

Οι τιμές των φυσικών παραμέτρων της κυκλικής κίνησης υπολογιστήκαν βάση τις τιμές των μεταβλητών που δήλωσε ο χρήστης και με βάση τις μαθηματικές σχέσεις που δηλώθηκαν στο πρόγραμμα :

$$R = \frac{v}{\omega} \Rightarrow \text{ακτίνα τροχιάς}$$

$$v = \omega R \Rightarrow \text{γραμμική ταχύτητα}$$

$$\omega = \frac{v}{R} \Rightarrow \text{γωνιακή ταχύτητα}$$

$$F_{\kappa} = m \cdot \frac{v^2}{R} \Rightarrow \text{κεντρομόλος δύναμη}$$

`a1=isnan(R); %η εντολή isnan (is not a number) επιστρέφει true or false`

`a2=isnan(omega);`

`a3=isnan(v);`

`if (a1==1)`

`R=v/omega;`

`end`

`if (a2==1)`

`omega=v/R;`

`end`

`if (a3==1)`

`v=omega*R;`

`end`

`Fk=m*v*v/R;`

Στην περίπτωση τις πιο πάνω εφαρμογής η κεντρομόλος δύναμη που αναγκάζει τη σφαίρα να κινηθεί σε κυκλική τροχιά είναι η τάση του νήματος.

$$\text{Τάση νήματος} = F_{\kappa} = m \cdot \frac{v^2}{R} \Rightarrow \text{κεντρομόλος δύναμη}$$

Αν η σφαίρα περιφέρεται με ολοένα αυξανόμενη ταχύτητα, τότε σύμφωνα με τη πιο πάνω σχέση απαιτείται μεγαλύτερη δύναμη για να τη συγκρατήσει σε κυκλική τροχιά.

Αν η δύναμη αυτή υπερβεί την τάση θραύσης του νήματος, τότε αυτό κόβεται και η σφαίρα κινείται ευθύγραμμα κατά την εφαπτομένη της τροχιάς στη θέση που κόπηκε το νήμα.

Στην περίπτωση της εφαρμογής αυτής η τάση θραύσης του νήματος ορίστηκε 200 kgm/sec².

Όταν η τιμή της κεντρομόλου είναι μεγαλύτερη από 200 kgm/sec² τότε επιτρέπουμε στο σώμα να εκτελέσει την κυκλική κίνηση για μια περίοδο και τότε δείχνουμε την πορεία που ακολουθεί το σώμα μετά το σπάσιμο του νήματος, το οποίο κινείται ευθύγραμμα κατά την εφαπτομένη του τροχιάς στη θέση που κόπηκε το νήμα.

Η πορεία εκτέλεσης του προγράμματος με βάση την τιμή της κεντρομόλου δύναμης, έγινε με την χρήση της δομής *if — else — end*, όπου:

```
if (Fk<=200)
```

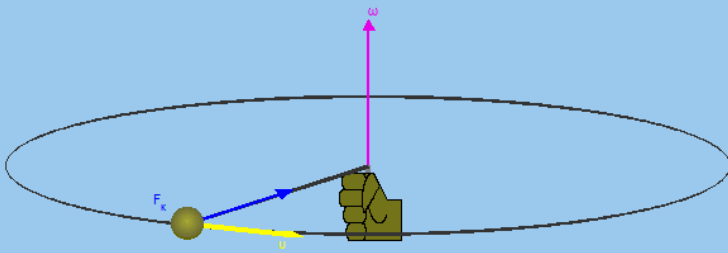
```
    %εντολές
```

```
else
```

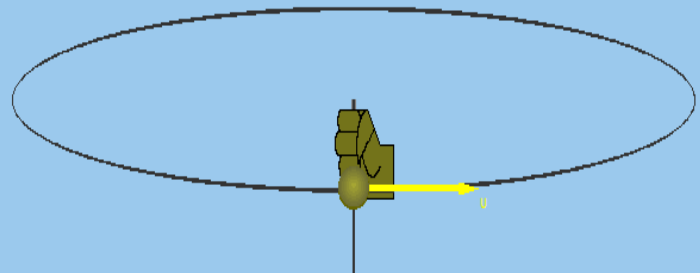
```
    %εντολές
```

```
end
```

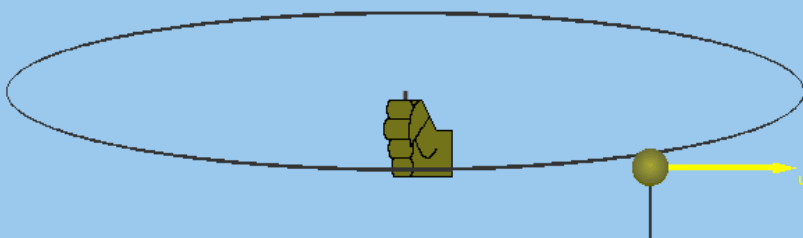
ΔΥΝΑΜΙΚΗ ΣΤΟ ΕΠΙΠΕΔΟ
Ομαλή κυκλική κίνηση - Κεντρομόλος δύναμη



ΔΥΝΑΜΙΚΗ ΣΤΟ ΕΠΙΠΕΔΟ
Ομαλή κυκλική κίνηση - Κεντρομόλος δύναμη



ΔΥΝΑΜΙΚΗ ΣΤΟ ΕΠΙΠΕΔΟ
Ομαλή κυκλική κίνηση - Κεντρομόλος δύναμη



$m = 3$ Kgm
 $R = 2$ m
 $u = 20$ m/sec
 $\omega = 10$ rad/sec
 $F_k = 600$ Kgm/sec²

$t = 0.04$ sec

$$F_k = \frac{mu^2}{R} \quad (1)$$

$$u = \omega R \quad (2)$$

$$F_k = m\omega^2 R \quad (3)$$

$$F_k = m\omega u \quad (4)$$

Αν η κεντρομόλος δύναμη γίνει μεγαλύτερη από την τάση θραύσης του νήματος τότε αυτό κόβεται και η σφαίρα κινείται ευθύγραμμα κατά τη διεύθυνση της γραμμικής ταχύτητας

- να παρακολουθεί τη πορεία του σώματος και των δυνάμεων που ασκούνται σε αυτό.

Για να επιτύχουμε το πιο πάνω σχεδιάγραμμα που απεικονίζει την κίνηση του σώματος, το οποίο είναι στερεωμένο σε νήμα και εκτελεί κυκλική κίνηση με την βοήθεια ενός χεριού, χρησιμοποιήσαμε την εντολή *fill* όπου δέχεται συντεταμένες (X,Y) και δημιουργεί κλειστά πολύγωνα με καθορισμένο χρώμα.

ΠΑΡΑΡΤΗΜΑ

A. Κώδικας MATLAB

```
function varargout = kentromolos(varargin)
% KENTROMOLOS M-file for kentromolos.fig
%   KENTROMOLOS, by itself, creates a new KENTROMOLOS or raises the existing
%   singleton*.
%
%   H = KENTROMOLOS returns the handle to a new KENTROMOLOS or the handle to
%   the existing singleton*.
%
%   KENTROMOLOS('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in KENTROMOLOS.M with the given input arguments.
%
%   KENTROMOLOS('Property','Value',...) creates a new KENTROMOLOS or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before kentromolos_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to kentromolos_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help kentromolos

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @kentromolos_OpeningFcn, ...
                  'gui_OutputFcn',  @kentromolos_OutputFcn, ...
                  'gui_LayoutFcn',   [] , ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before kentromolos is made visible.
function kentromolos_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to kentromolos (see VARARGIN)

% Choose default command line output for kentromolos
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes kentromolos wait for user response (see UIRESUME)
% uiwait(handles.figure1);

% --- Outputs from this function are returned to the command line.
function varargout = kentromolos_OutputFcn(hObject, eventdata, handles)
% varargout  cell array for returning output args (see VARARGOUT);
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

function edit1_Callback(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
```

```

% Hints: get(hObject,'String') returns contents of edit1 as text
%         str2double(get(hObject,'String')) returns contents of edit1 as a double

% --- Executes during object creation, after setting all properties.
function edit1_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit2_Callback(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit2 as text
%         str2double(get(hObject,'String')) returns contents of edit2 as a double

% --- Executes during object creation, after setting all properties.
function edit2_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit3_Callback(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit3 as text
%         str2double(get(hObject,'String')) returns contents of edit3 as a double

% --- Executes during object creation, after setting all properties.
function edit3_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit4_Callback(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit4 as text
%         str2double(get(hObject,'String')) returns contents of edit4 as a double

% --- Executes during object creation, after setting all properties.
function edit4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit5_Callback(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

```

```

% Hints: get(hObject,'String') returns contents of edit5 as text
%         str2double(get(hObject,'String')) returns contents of edit5 as a double

% --- Executes during object creation, after setting all properties.
function edit5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

function edit6_Callback(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'String') returns contents of edit6 as text
%         str2double(get(hObject,'String')) returns contents of edit6 as a double

% --- Executes during object creation, after setting all properties.
function edit6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to edit6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: edit controls usually have a white background on Windows.
%         See ISPC and COMPUTER.
if ispc && isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor','white');
end

% --- Executes on button press in pushbutton1.
function pushbutton1_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global m;
global v;
global omega;
global RR;
global t;
global ryt;

m=str2double(get(handles.edit1,'String')); %object mass
RR=str2double(get(handles.edit2,'String')); %orbit radius
omega=str2double(get(handles.edit3,'String')); %angular velocity
v=str2double(get(handles.edit4,'String')); %linear velocity
t=str2double(get(handles.edit5,'String')); %observation time
ryt=str2double(get(handles.edit6,'String')); %simulation speed

global stam;

stam=0;
set(handles.pushbutton2,'string','Διακοπή');

global status;
status=0;

%check if RR, omega and v are specified
a1=isnan(RR);
a2=isnan(omega);
a3=isnan(v);
suma=a1+a2+a3;

%check variable values and issue warnings if values are outside of
%specified limits

if (suma~=1)
    hfin=warndlg('Επιλέξτε ακριβώς 2 από τις τρεις μεταβλητές');
    return
end

if (ryt>0.5|ryt<0.001)
    h=warndlg('Βάλτε στο Ρυθμό Προσομοίωσης τιμή μεταξύ 0.001 και 0.5');
    return
end

```



```

if (t<10)
    hfin=warndlg('Βάλτε στο Χρόνο Παρατήρησης τιμή μεγαλύτερη ή ίση του 10');
    return
end

if (t>400)
    hfin=warndlg('Βάλτε στο Χρόνο Παρατήρησης τιμή μικρότερη ή ίση του 400');
    return
end

if (m<1)
    hfin=warndlg('Βάλτε στη Μάζα τιμή μεγαλύτερη ή ίση του 1');
    return
end

if (m>3)
    hfin=warndlg('Βάλτε στη Μάζα τιμή μικρότερη ή ίση του 3');
    return
end

if (a1==0)

if (RR<0.5)
    hfin=warndlg('Βάλτε στην Ακτίνα Τροχιάς τιμή μεγαλύτερη ή ίση του 0.5');
    return
end

if (RR>2)
    hfin=warndlg('Βάλτε στην Ακτίνα Τροχιάς τιμή μικρότερη ή ίση του 2');
    return
end

end

if (a2==0)

if (omega<5)
    hfin=warndlg('Βάλτε στην Γωνιακή Ταχύτητα τιμή μεγαλύτερη ή ίση του 5');
    return
end

if (omega>10)
    hfin=warndlg('Βάλτε στην Γωνιακή Ταχύτητα τιμή μικρότερη ή ίση του 10');
    return
end

end

if (a3==0)

if (v<5)
    hfin=warndlg('Βάλτε στη Γραμμική Ταχύτητα τιμή μεγαλύτερη ή ίση του 5');
    return
end

if (v>10)
    hfin=warndlg('Βάλτε στη Γραμμική Ταχύτητα τιμή μικρότερη ή ίση του 10');
    return
end

end

ryte =-ryt+0.5+0.001;
set(handles.edit1,'enable','off');
set(handles.edit2,'enable','off');
set(handles.edit3,'enable','off');
set(handles.edit4,'enable','off');
set(handles.edit5,'enable','off');
set(handles.edit6,'enable','off');

set(handles.pushbutton1,'enable','off');
set(handles.pushbutton3,'enable','off');
set(handles.pushbutton4,'enable','off');

axes(handles.axes1)
axis off;

```

```

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

```

```

%set some variables
L0=1;
th=0:0.01:pi;
th1=0:0.01:2*pi;
Rmin=0.0125;
Rmin1=0.04;
L=0.5;
Dx=0.325;
Dy=-0.025;

%set initial co-ordinates of objects on the diagram
%wall co-ordinates
xtoixos=[0,3,3,0];
ytoixos=[-0.5,-0.5,0.5,0.5];

%hand co-ordinates
xxeri=[1.00,...
        1.10,...
        1.15,...
        1.20,...
        1.20,...
        1.05,...
        1.05,...
        1.015+Rmin*cos(3*pi/2-th),...
        1.005+Rmin*cos(3*pi/2-th),...
        0.99+Rmin*cos(3*pi/2-th),...
        1.00+Rmin*cos(3*pi/2-th)];

xxeri=xxeri+L-0.06*L-0.5+0.03;

yxeri=[0.40,...
        0.40,...
        0.36,...
        0.36,...
        0.30,...
        0.30,...
        0.30,...
        0.30+Rmin+Rmin*sin(3*pi/2-th),...
        0.325+Rmin+Rmin*sin(3*pi/2-th),...
        0.350+Rmin+Rmin*sin(3*pi/2-th),...
        0.375+Rmin+Rmin*sin(3*pi/2-th)];

yxeri=yxeri-0.35+0.13+0.015;

xxeri1=[1.015+Rmin*cos(3*pi/2-th),...
         1.015,1.060,...
         1.060+Rmin*cos(pi/2-th),1.015];

xxeri1=xxeri1+L-0.06*L-0.5+0.03;

yxeri1=[0.30+Rmin+Rmin*sin(3*pi/2-th),...
         0.325,0.325,...
         0.30+Rmin+Rmin*sin(pi/2-th),0.30];
yxeri1=yxeri1-0.35+0.13+0.015;

xxeri2=[1.005+Rmin*cos(3*pi/2-th),...
         1.005,1.060,...
         1.060+Rmin*cos(pi/2-th),1.005];

xxeri2=xxeri2+L-0.06*L-0.5+0.03;

yxeri2=[0.325+Rmin+Rmin*sin(3*pi/2-th),...
         0.350,0.350,...
         0.325+Rmin+Rmin*sin(pi/2-th),0.325];
yxeri2=yxeri2-0.35+0.13+0.015;

xxeri3=[0.99+Rmin*cos(3*pi/2-th),...
         0.99,1.060,...
         1.060+Rmin*cos(pi/2-th),0.99];

xxeri3=xxeri3+L-0.06*L-0.5+0.03;

yxeri3=[0.350+Rmin+Rmin*sin(3*pi/2-th),...
         0.375,0.375,...
         0.350+Rmin+Rmin*sin(pi/2-th),0.350];
yxeri3=yxeri3-0.35+0.13+0.015;

xxeri4=[1.00+Rmin*cos(3*pi/2-th),...
         1.00,1.060,...
         1.060+Rmin*cos(pi/2-th),1.00];
xxeri4=xxeri4+L-0.06*L-0.5+0.03;

```

```

yxeri4=[0.375+Rmin+Rmin*sin(3*pi/2-th),...
        0.40,0.40,...
        0.375+Rmin+Rmin*sin(pi/2-th),0.375];
yxeri4=yxeri4-0.35+0.13+0.015;

xxeri5=[1.10+Rmin1*cos(pi/2+2*th/3),...
        1.10+Rmin1*cos(pi/2+2*pi/3)+3*Rmin*sin(th/2),1.125];
xxeri5=xxeri5+L-0.06*L-0.5+0.03;
yxeri5=[0.40-Rmin1+Rmin1*sin(pi/2+2*th/3),...
        0.40-Rmin1+Rmin1*sin(pi/2+2*pi/3)-2*Rmin*cos(pi/2+th/2),0.38];
yxeri5=yxeri5-0.35+0.13+0.015;

xxeri6=[0.1,0.15+Rmin1*cos(5*pi/4+th/2),0.15+Rmin1*cos(5*pi/4+pi/2-th/2)];
xxeri6=xxeri6+L-0.06*L+0.50;
yxeri6=[0.0,Rmin1*sin(5*pi/4+th/2),Rmin1*sin(5*pi/4+pi/2-th/2)];
yxeri6=yxeri6+0.015+0.13+0.015;

%bar co-ordinates
xstick=[1.4,1.42,1.42,1.4]-0.035;
ystick=[0.185,0.185,0.2,0.2]-0.024;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%5<w<10
%1<m<3
%t=10;
%ryt=0.05;
%m=3;
%omega=5;
%v=5;
%RR=v/omega;

%if not specified set the radius of rotation
if (a1==1)
    RR=v/omega;
end

%if not specified set the angular velocity
if (a2==1)
    omega=v/RR;
end

%if not specified set the linear velocity
if (a3==1)
    v=omega*RR;
end

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%force= mass * (linear velocity)^2 / radius;
Fk=m*v*v/RR;

%Lom = 0.1 * (angular velocity) / 5
Lom=0.1*omega/5;

a=0.65*RR;
b=0.05*RR;
%pythagoras theorem for ab
ab=(a*a+b*b)^0.5;
xc=1.41-0.035;
yc=0.2-0.024;
Rb=0.06*(m/3)^(1/3);

LFk=0.5;
LFkx=LFk*a;
LFky=LFk*b;

Lv=0.4;
Lvx=Lv*a;
Lvy=Lv*b;

%estimate orbit in x-direction: a*cosine(theta1), same but shifted by 0.005
xtroxia=[a*cos(th1), (a+0.005)*cos(th1)]+xc;
%estimate orbit in y-direction: a*sine(theta1), same but shifted by 0.005
ytroxia=[b*sin(th1), (b+0.005)*sin(th1)]+yc;

xvectorom=xc+[0.0065,0.0065,0.025,0.0,-0.025,-0.0065,-0.0065];
yvectorom=yc+[0.0,Lom,Lom,Lom+0.02,Lom,Lom,0.0];

xpivakas=[0.2,2.75,2.75,0.2];
ypivakas=[-0.45 -0.45 ,-0.05,-0.05];

```

```

xpivakasb=[0.18,2.77,2.77,0.18];
ypivakasb=[-0.46 -0.46 , -0.04,-0.04];

xgom=xc;
ygom=yc+Lom+0.03;

nm=num2str(0.01*round(100*m));
nv=num2str(0.01*round(100*v));
nR=num2str(0.01*round(100*RR));
nom=num2str(0.01*round(100*omega));
nfk=num2str(0.01*round(100*Fk));

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
t1=(3*pi/2)/omega;
NN=t1/0.05;
NN=round(NN);
%time step for simulation if Force > 200N
Dt=t1/NN;

%if the force is smaller than 200N, execute this loop; else execute loop

if (Fk<=200)
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%simulate system for t seconds in time steps of 0.05
for time=0.0:0.05:t

    if (stam==1)
        ccl=stam;
        while (ccl==1)
            ccl=stam;
            pause(0.01);
            if (status==1)
                return
            end
        end
    end
end

    nt=num2str(0.01*round(100*time));
    phi1=omega*time; %calculate angle at each time step

xhand=(a/20)*cos(phi1);
yhand=(b/20)*sin(phi1);

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%co-ordinates for hand image
xxeri=[1.00,...
        1.10,...
        1.15,...
        1.20,...
        1.20,...
        1.05,...
        1.05,...
        1.015+Rmin*cos(3*pi/2-th),...
        1.005+Rmin*cos(3*pi/2-th),...
        0.99+Rmin*cos(3*pi/2-th),...
        1.00+Rmin*cos(3*pi/2-th)];

xxeri=xxeri+L-0.06*L-0.5+0.03+xhand;

yxeri=[0.40,...
        0.40,...
        0.36,...
        0.36,...
        0.30,...
        0.30,...
        0.30,...
        0.30+Rmin+Rmin*sin(3*pi/2-th),...
        0.325+Rmin+Rmin*sin(3*pi/2-th),...
        0.350+Rmin+Rmin*sin(3*pi/2-th),...
        0.375+Rmin+Rmin*sin(3*pi/2-th)];

yxeri=yxeri-0.35+0.13+0.015+yhand;

xxeri1=[1.015+Rmin*cos(3*pi/2-th),...
        1.015,1.060,...

```

```

1.060+Rmin*cos(pi/2-th),1.015];

xxeri1=xxeri1+L-0.06*L-0.5+0.03+xhand;

yxeri1=[0.30+Rmin+Rmin*sin(3*pi/2-th),...
0.325,0.325,...
0.30+Rmin+Rmin*sin(pi/2-th),0.30];
yxeri1=yxeri1-0.35+0.13+0.015+yhand;

xxeri2=[1.005+Rmin*cos(3*pi/2-th),...
1.005,1.060,...
1.060+Rmin*cos(pi/2-th),1.005];

xxeri2=xxeri2+L-0.06*L-0.5+0.03+xhand;

yxeri2=[0.325+Rmin+Rmin*sin(3*pi/2-th),...
0.350,0.350,...
0.325+Rmin+Rmin*sin(pi/2-th),0.325];
yxeri2=yxeri2-0.35+0.13+0.015+yhand;

xxeri3=[0.99+Rmin*cos(3*pi/2-th),...
0.99,1.060,...
1.060+Rmin*cos(pi/2-th),0.99];

xxeri3=xxeri3+L-0.06*L-0.5+0.03+xhand;

yxeri3=[0.350+Rmin+Rmin*sin(3*pi/2-th),...
0.375,0.375,...
0.350+Rmin+Rmin*sin(pi/2-th),0.350];
yxeri3=yxeri3-0.35+0.13+0.015+yhand;

xxeri4=[1.00+Rmin*cos(3*pi/2-th),...
1.00,1.060,...
1.060+Rmin*cos(pi/2-th),1.00];

xxeri4=xxeri4+L-0.06*L-0.5+0.03+xhand;

yxeri4=[0.375+Rmin+Rmin*sin(3*pi/2-th),...
0.40,0.40,...
0.375+Rmin+Rmin*sin(pi/2-th),0.375];
yxeri4=yxeri4-0.35+0.13+0.015+yhand;

xxeri5=[1.10+Rmin1*cos(pi/2+2*th/3),...
1.10+Rmin1*cos(pi/2+2*pi/3)+3*Rmin*sin(th/2),1.125];
xxeri5=xxeri5+L-0.06*L-0.5+0.03+xhand;
yxeri5=[0.40-Rmin1+Rmin1*sin(pi/2+2*th/3),...
0.40-Rmin1+Rmin1*sin(pi/2+2*pi/3)-2*Rmin*cos(pi/2+th/2),0.38];
yxeri5=yxeri5-0.35+0.13+0.015+yhand;

xxeri6=[0.1,0.15+Rmin1*cos(5*pi/4+th/2),0.15+Rmin1*cos(5*pi/4+pi/2-th/2)];
xxeri6=xxeri6+L-0.06*L+0.50+xhand;
yxeri6=[0.0,Rmin1*sin(5*pi/4+th/2),Rmin1*sin(5*pi/4+pi/2-th/2)];
yxeri6=yxeri6+0.015+0.13+0.015+yhand;
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

%co-ordinates for bar

xbar=xc+[0.005*sin(phi1),0.005*sin(phi1)+a*cos(phi1),...
-0.005*sin(phi1)+a*cos(phi1),-0.005*sin(phi1)];
ybar=yc+[-0.005*cos(phi1),-0.005*cos(phi1)+b*sin(phi1),...
0.005*cos(phi1)+b*sin(phi1),0.005*cos(phi1)];

xbody=xc+a*cos(phi1)+Rb*cos(th1);
ybody=yc+b*sin(phi1)+(Rb-0.6*Rb)*sin(th1);

xbody1=xc+a*cos(phi1)+0.97*Rb*cos(th1);
ybody1=yc+b*sin(phi1)+0.97*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody2=xc+a*cos(phi1)+0.93*Rb*cos(th1);
ybody2=yc+b*sin(phi1)+0.93*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody3=xc+a*cos(phi1)+0.85*Rb*cos(th1);
ybody3=yc+b*sin(phi1)+0.85*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody4=xc+a*cos(phi1)+0.75*Rb*cos(th1);
ybody4=yc+b*sin(phi1)+0.75*(Rb-0.6*Rb)*sin(th1)+0.0025;

```

```

xbody5=xc+a*cos(phi1)+0.62*Rb*cos(th1);
ybody5=yc+b*sin(phi1)+0.62*(Rb-0.6*Rb)*sin(th1)+0.0025;

xbody6=xc+a*cos(phi1)+0.48*Rb*cos(th1);
ybody6=yc+b*sin(phi1)+0.48*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody7=xc+a*cos(phi1)+0.32*Rb*cos(th1);
ybody7=yc+b*sin(phi1)+0.32*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody8=xc+a*cos(phi1)+0.14*Rb*cos(th1);
ybody8=yc+b*sin(phi1)+0.14*(Rb-0.6*Rb)*sin(th1)+0.0035;

xbody9=xc+a*cos(phi1)+0.02*Rb*cos(th1);
ybody9=yc+b*sin(phi1)+0.02*(Rb-0.025)*sin(th1)+0.0035;

xk=xc+a*cos(phi1);
yk=yc+b*sin(phi1);

xvectorF=xk+[0.005*sin(phi1),0.005*sin(phi1)-LFkx*cos(phi1),...
0.005*sin(phi1)-LFkx*cos(phi1)+0.01*sin(phi1),...
-(LFkx+0.085*a)*cos(phi1),...
-0.005*sin(phi1)-LFkx*cos(phi1)-0.01*sin(phi1),...
-0.005*sin(phi1)-LFkx*cos(phi1),-0.005*sin(phi1)];
yvectorF=yk+[-0.005*cos(phi1),-0.005*cos(phi1)-LFky*sin(phi1),...
-0.005*cos(phi1)-LFky*sin(phi1)-0.01*cos(phi1),...
-(LFky+0.085*b)*sin(phi1),...
0.005*cos(phi1)-LFky*sin(phi1)+0.01*cos(phi1),...
0.005*cos(phi1)-LFky*sin(phi1),0.005*cos(phi1)];

xgF=xc+0.8*a*cos(phi1-pi/10);
ygF=yc+0.8*b*sin(phi1-pi/10);

xgv=xc+1.2*a*cos(phi1+pi/10);
ygv=yc+1.2*b*sin(phi1+pi/10);

xvectorv=xk+[0.004*cos(phi1),0.004*cos(phi1)-Lvz*sin(phi1)/ab,...
0.004*cos(phi1)-Lvz*sin(phi1)/ab+0.02*a*cos(phi1)/ab,...
-(Lvz+0.085*a)*sin(phi1)/ab,...
-0.004*cos(phi1)-Lvz*sin(phi1)/ab-0.02*a*cos(phi1)/ab,...
-0.004*cos(phi1)-Lvz*sin(phi1)/ab,-0.004*cos(phi1)];
yvectorv=yk+[0.004*sin(phi1),0.004*sin(phi1)+Lvy*cos(phi1)/ab,...
0.004*sin(phi1)+Lvy*cos(phi1)/ab+0.02*b*sin(phi1)/ab,...
(Lvy+0.085*b)*cos(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab-0.02*b*sin(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab,-0.004*sin(phi1)];

%start drawing image

fill(xtoixos,ytoixos,[0.6078 0.7882 0.9294],...
xpivakasb,ypivakasb,[0.5,0.4,0.3],...
xpivakas,ypivakas,[0.95,0.95,0.95],...
xstick,ystick,[0.2,0.2,0.2],...
xtroxia,ytroxia,[0.2,0.2,0.2],...
xvectorF,yvectorF,[0,0,0,1],...
xvectorv,yvectorv,[1,1,0],...
xvectorom,yvectorom,[0.9,0.9,0.9],...
xbody,ybody,[0.4,0.4,0.4],...
xbody1,ybody1,[0.42,0.42,0.2],...
xbody2,ybody2,[0.44,0.44,0.2],...
xbody3,ybody3,[0.46,0.46,0.2],...
xbody4,ybody4,[0.49,0.49,0.2],...
xbody5,ybody5,[0.52,0.52,0.2],...
xbody6,ybody6,[0.56,0.56,0.2],...
xbody7,ybody7,[0.6,0.6,0.2],...
xbody8,ybody8,[0.65,0.65,0.2],...
xbody9,ybody9,[0.7,0.7,0.2],...
'LineStyle','None')

hold on

%draw hand
fill(xxeri+Dx,yxeri+Dy,[0.45,0.45,0.1],...
xxeri6+Dx,yxeri6+Dy,[0.45,0.45,0.1],...
xxeri1+Dx,yxeri1+Dy,[0.45,0.45,0.1],...
xxeri2+Dx,yxeri2+Dy,[0.45,0.45,0.1],...
xxeri3+Dx,yxeri3+Dy,[0.45,0.45,0.1],...
xxeri4+Dx,yxeri4+Dy,[0.45,0.45,0.1],...
xxeri5+Dx,yxeri5+Dy,[0.45,0.45,0.1])

```

```

hold on

%draw orbit and object body
fill(xtroxia,ytroxia,[0.2,0.2,0.2],...
     xbar,ybar,[0.2,0.2,0.2],...
     xvectorF,yvectorF,[0,0.0,1],...
     xvectorv,yvectorv,[1,1,0],...
     xvectorom,yvectorom,[0.9,0.,0.9],...
     xbody,ybody,[0.4,0.4,0.],...
     xbody1,ybody1,[0.42,0.42,0.2],...
     xbody2,ybody2,[0.44,0.44,0.2],...
     xbody3,ybody3,[0.46,0.46,0.2],...
     xbody4,ybody4,[0.49,0.49,0.2],...
     xbody5,ybody5,[0.52,0.52,0.2],...
     xbody6,ybody6,[0.56,0.56,0.2],...
     xbody7,ybody7,[0.6,0.6,0.2],...
     xbody8,ybody8,[0.65,0.65,0.2],...
     xbody9,ybody9,[0.7,0.7,0.2],...
     'Linestyle','None')

hold off
% text(0.8,-0.07,'ΟΜΑΔΗ ΚΥΚΛΙΚΗ ΚΙΝΗΣΗ - Κεντρομόλος Δύναμη ','Color','b','FontSize',9)
% text(1.75,-0.07,'Γραμμική Ταχύτητα','Color','k','FontSize',8)

%fill in table with variables, their values, units and equations
text(xgF,ygF,'F_k','Color','b','FontSize',8)
text(xgv,ygv,'v','Color','y','FontSize',8)
text(xgom,ygom,'ω','Color',[0.9 0 0.9],'FontSize',8)

text(0.3,-0.15+0.05,'m =','Color','k','FontSize',8)
text(0.4,-0.15+0.05,nm,'Color','k','FontSize',8)
text(0.55,-0.15+0.05,'Kgm','Color','k','FontSize',8)

text(0.3,-0.2+0.05,'R =','Color','k','FontSize',8)
text(0.4,-0.2+0.05,nR,'Color','k','FontSize',8)
text(0.55,-0.2+0.05,'m','Color','k','FontSize',8)

text(0.3,-0.25+0.05,'v =','Color','k','FontSize',8)
text(0.4,-0.25+0.05,nv,'Color','k','FontSize',8)
text(0.55,-0.25+0.05,'m/sec','Color','k','FontSize',8)

text(0.3,-0.3+0.05,'ω =','Color','k','FontSize',8)
text(0.4,-0.3+0.05,nom,'Color','k','FontSize',8)
text(0.55,-0.3+0.05,'rad/sec','Color','k','FontSize',8)

text(0.3,-0.35+0.05,'F_k =','Color','k','FontSize',8)
text(0.4,-0.35+0.05,nfk,'Color','k','FontSize',8)
text(0.55,-0.35+0.05,'Kgm/sec^2','Color','k','FontSize',8)

text(0.3+1,-0.15+0.05,'t =','Color','b','FontSize',8)
text(0.4+1,-0.15+0.05,nt,'Color','b','FontSize',8)
text(0.55+1,-0.15+0.05,'sec','Color','b','FontSize',8)

axis([0,3,-0.5,0.5])
axis off

arx1 = [0.85, 0.95]+1.64-0.05;
tell1 = [-0.12, -0.12]-0.03+0.05;

line(arx1, tell1, 'Color','r');

text(0.6+1.9-0.05,-0.1-0.03+0.05,'mv^2','Color','r','FontSize',9)
text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045 , 'F_k = (1)','Color','r','FontSize',9)
text(0.62+1.9-0.05 , -0.14-0.03+0.05, 'R ', 'Color','r','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.06 , 'v = ωR (2)','Color','b','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.12 , 'F_k = mω^2R (3)','Color','r','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.18 , 'F_k = mωv (4)','Color','r','FontSize',9)
pause(ryte)

end

%if Force is >200N, execute this loop which shows that the object escapes
%the orbit

```

```

#####
else
#####
% pause(1000)
for time=0.0:Dt:t1

    if (stam==1)
        ccl=stam;
        while (ccl==1)
            ccl=stam;
            pause(0.01);
            if (status==1)
                return
            end
        end
    end
end

    nt=num2str(0.01*round(100*time));
    %estimate angle of rotation
    phil=omega*time;

    xhand=(a/20)*cos(phil);
    yhand=(b/20)*sin(phil);

#####
xxeri=[1.00,...
        1.10,...
        1.15,...
        1.20,...
        1.20,...
        1.05,...
        1.05,...
        1.015+Rmin*cos(3*pi/2-th),...
        1.005+Rmin*cos(3*pi/2-th),...
        0.99+Rmin*cos(3*pi/2-th),...
        1.00+Rmin*cos(3*pi/2-th)];

    xxeri=xxeri+L-0.06*L-0.5+0.03+xhand;

    yxeri=[0.40,...
           0.40,...
           0.36,...
           0.36,...
           0.30,...
           0.30,...
           0.30,...
           0.30+Rmin+Rmin*sin(3*pi/2-th),...
           0.325+Rmin+Rmin*sin(3*pi/2-th),...
           0.350+Rmin+Rmin*sin(3*pi/2-th),...
           0.375+Rmin+Rmin*sin(3*pi/2-th)];

    yxeri=yxeri-0.35+0.13+0.015+yhand;

    xxeri1=[1.015+Rmin*cos(3*pi/2-th),...
            1.015,1.060,...
            1.060+Rmin*cos(pi/2-th),1.015];

    xxeri1=xxeri1+L-0.06*L-0.5+0.03+xhand;

    yxeri1=[0.30+Rmin+Rmin*sin(3*pi/2-th),...
            0.325,0.325,...
            0.30+Rmin+Rmin*sin(pi/2-th),0.30];
    yxeri1=yxeri1-0.35+0.13+0.015+yhand;

    xxeri2=[1.005+Rmin*cos(3*pi/2-th),...
            1.005,1.060,...
            1.060+Rmin*cos(pi/2-th),1.005];

    xxeri2=xxeri2+L-0.06*L-0.5+0.03+xhand;

    yxeri2=[0.325+Rmin+Rmin*sin(3*pi/2-th),...
            0.350,0.350,...
            0.325+Rmin+Rmin*sin(pi/2-th),0.325];
    yxeri2=yxeri2-0.35+0.13+0.015+yhand;

    xxeri3=[0.99+Rmin*cos(3*pi/2-th),...
            0.99,1.060,...
            1.060+Rmin*cos(pi/2-th),0.99];

```



```

xxeri3=xxeri3+L-0.06*L-0.5+0.03+xhand;

yxeri3=[0.350+Rmin+Rmin*sin(3*pi/2-th),...
        0.375,0.375,...
        0.350+Rmin+Rmin*sin(pi/2-th),0.350];
yxeri3=yxeri3-0.35+0.13+0.015+yhand;

xxeri4=[1.00+Rmin*cos(3*pi/2-th),...
        1.00,1.060,...
        1.060+Rmin*cos(pi/2-th),1.00];

xxeri4=xxeri4+L-0.06*L-0.5+0.03+xhand;

yxeri4=[0.375+Rmin+Rmin*sin(3*pi/2-th),...
        0.40,0.40,...
        0.375+Rmin+Rmin*sin(pi/2-th),0.375];
yxeri4=yxeri4-0.35+0.13+0.015+yhand;

xxeri5=[1.10+Rmin1*cos(pi/2+2*th/3),...
        1.10+Rmin1*cos(pi/2+2*pi/3)+3*Rmin*sin(th/2),1.125];
xxeri5=xxeri5+L-0.06*L-0.5+0.03+xhand;
yxeri5=[0.40-Rmin1+Rmin1*sin(pi/2+2*th/3),...
        0.40-Rmin1+Rmin1*sin(pi/2+2*pi/3)-2*Rmin*cos(pi/2+th/2),0.38];
yxeri5=yxeri5-0.35+0.13+0.015+yhand;

xxeri6=[0.1,0.15+Rmin1*cos(5*pi/4+th/2),0.15+Rmin1*cos(5*pi/4+pi/2-th/2)];
xxeri6=xxeri6+L-0.06*L+0.50+xhand;
yxeri6=[0.0,Rmin1*sin(5*pi/4+th/2),Rmin1*sin(5*pi/4+pi/2-th/2)];
yxeri6=yxeri6+0.015+0.13+0.015+yhand;

%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

xbar=xc+[0.005*sin(phi1),0.005*sin(phi1)+a*cos(phi1),...
        -0.005*sin(phi1)+a*cos(phi1),-0.005*sin(phi1)];
ybar=yc+[-0.005*cos(phi1),-0.005*cos(phi1)+b*sin(phi1),...
        0.005*cos(phi1)+b*sin(phi1),0.005*cos(phi1)];

xbody=xc+a*cos(phi1)+Rb*cos(th1);
ybody=yc+b*sin(phi1)+(Rb-0.6*Rb)*sin(th1);

xbody1=xc+a*cos(phi1)+0.97*Rb*cos(th1);
ybody1=yc+b*sin(phi1)+0.97*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody2=xc+a*cos(phi1)+0.93*Rb*cos(th1);
ybody2=yc+b*sin(phi1)+0.93*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody3=xc+a*cos(phi1)+0.85*Rb*cos(th1);
ybody3=yc+b*sin(phi1)+0.85*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody4=xc+a*cos(phi1)+0.75*Rb*cos(th1);
ybody4=yc+b*sin(phi1)+0.75*(Rb-0.6*Rb)*sin(th1)+0.0025;

xbody5=xc+a*cos(phi1)+0.62*Rb*cos(th1);
ybody5=yc+b*sin(phi1)+0.62*(Rb-0.6*Rb)*sin(th1)+0.0025;

xbody6=xc+a*cos(phi1)+0.48*Rb*cos(th1);
ybody6=yc+b*sin(phi1)+0.48*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody7=xc+a*cos(phi1)+0.32*Rb*cos(th1);
ybody7=yc+b*sin(phi1)+0.32*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody8=xc+a*cos(phi1)+0.14*Rb*cos(th1);
ybody8=yc+b*sin(phi1)+0.14*(Rb-0.6*Rb)*sin(th1)+0.0035;

xbody9=xc+a*cos(phi1)+0.02*Rb*cos(th1);
ybody9=yc+b*sin(phi1)+0.02*(Rb-0.6*Rb)*sin(th1)+0.0035;

xk=xc+a*cos(phi1);
yk=yc+b*sin(phi1);

xvectorF=xk+[0.005*sin(phi1),0.005*sin(phi1)-LFkx*cos(phi1),...
            0.005*sin(phi1)-LFkx*cos(phi1)+0.01*sin(phi1),...
            -(LFkx+0.085*a)*cos(phi1),...
            -0.005*sin(phi1)-LFkx*cos(phi1)-0.01*sin(phi1),...
            -0.005*sin(phi1)-LFkx*cos(phi1),-0.005*sin(phi1)];
yvectorF=yk+[-0.005*cos(phi1),-0.005*cos(phi1)-LFky*sin(phi1),...
            -0.005*cos(phi1)-LFky*sin(phi1)-0.01*cos(phi1),...

```

```

-(LFky+0.085*b)*sin(phi1),...
0.005*cos(phi1)-LFky*sin(phi1)+0.01*cos(phi1),...
0.005*cos(phi1)-LFky*sin(phi1),0.005*cos(phi1)];

xgF=xc+0.8*a*cos(phi1-pi/10);
ygF=yc+0.8*b*sin(phi1-pi/10);

xgv=xc+1.2*a*cos(phi1+pi/10);
ygv=yc+1.2*b*sin(phi1+pi/10);

xvectorv=xk+[0.004*cos(phi1),0.004*cos(phi1)-Lvx*sin(phi1)/ab,...
0.004*cos(phi1)-Lvx*sin(phi1)/ab+0.04*a*cos(phi1)/ab,...
-(Lvx+0.085*a)*sin(phi1)/ab,...
-0.004*cos(phi1)-Lvx*sin(phi1)/ab-0.04*a*cos(phi1)/ab,...
-0.004*cos(phi1)-Lvx*sin(phi1)/ab,-0.004*cos(phi1)];
yvectorv=yk+[0.004*sin(phi1),0.004*sin(phi1)+Lvy*cos(phi1)/ab,...
0.004*sin(phi1)+Lvy*cos(phi1)/ab+0.04*b*sin(phi1)/ab,...
(Lvy+0.085*b)*cos(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab-0.04*b*sin(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab,-0.004*sin(phi1)];

fill(xtoixos,ytoixos,[0.6078 0.7882 0.9294],...
xpivakasb,ypivakasb,[0.5,0.4,0.3],...
xpivakas,ypivakas,[0.95,0.95,0.95],...
xstick,ystick,[0.5,0.5,0.5],...
xtroxia,ytroxia,[0.2,0.2,0.2],...
xvectorF,yvectorF,[0,0,0,1],...
xvectorv,yvectorv,[1.0,1.0,0.0],...
xvectorom,yvectorom,[0.9,1.0,0.9],...
xbody,ybody,[0.7,0.7,0.2],...
xbody,ybody,[0.4,0.4,0.],...
xbody1,ybody1,[0.42,0.42,0.2],...
xbody2,ybody2,[0.44,0.44,0.2],...
xbody3,ybody3,[0.46,0.46,0.2],...
xbody4,ybody4,[0.49,0.49,0.2],...
xbody5,ybody5,[0.52,0.52,0.2],...
xbody6,ybody6,[0.56,0.56,0.2],...
xbody7,ybody7,[0.6,0.6,0.2],...
xbody8,ybody8,[0.65,0.65,0.2],...
xbody9,ybody9,[0.7,0.7,0.2],...
'LineStyle','None')

hold on

fill(xxeri+Dx,yxeri+Dy,[0.45,0.45,0.1],...
xxeri6+Dx,yxeri6+Dy,[0.45,0.45,0.1],...
xxeri1+Dx,yxeri1+Dy,[0.45,0.45,0.1],...
xxeri2+Dx,yxeri2+Dy,[0.45,0.45,0.1],...
xxeri3+Dx,yxeri3+Dy,[0.45,0.45,0.1],...
xxeri4+Dx,yxeri4+Dy,[0.45,0.45,0.1],...
xxeri5+Dx,yxeri5+Dy,[0.45,0.45,0.1])

hold on

fill(xtroxia,ytroxia,[0.2,0.2,0.2],...
xbar,ybar,[0.2,0.2,0.2],...
xvectorF,yvectorF,[0,0,0,1],...
xvectorv,yvectorv,[1.0,1.0,0.0],...
xvectorom,yvectorom,[0.9,0.,0.9],...
xbody,ybody,[0.7,0.7,0.2],...
xbody,ybody,[0.4,0.4,0.],...
xbody1,ybody1,[0.42,0.42,0.2],...
xbody2,ybody2,[0.44,0.44,0.2],...
xbody3,ybody3,[0.46,0.46,0.2],...
xbody4,ybody4,[0.49,0.49,0.2],...
xbody5,ybody5,[0.52,0.52,0.2],...
xbody6,ybody6,[0.56,0.56,0.2],...
xbody7,ybody7,[0.6,0.6,0.2],...
xbody8,ybody8,[0.65,0.65,0.2],...
xbody9,ybody9,[0.7,0.7,0.2],...
'LineStyle','None')

hold off

% text(0.8,-0.07,'ΟΜΑΔΗ ΚΥΚΛΙΚΗ ΚΙΝΗΣΗ - Κεντρομόλος Δύναμη ','Color','b','FontSize',9)
% text(1.75,-0.07,'Γραμμική Ταχύτητα','Color','k','FontSize',8)

text(xgF,ygF,'F_k','Color','b','FontSize',8)

```

```

text(xgv,ygv,'v','Color','y','FontSize',8)
text(xgom,ygom,'ω','Color',[0.9 0 0.9],'FontSize',8)

text(0.3,-0.15+0.05,'m =','Color','k','FontSize',8)
text(0.4,-0.15+0.05,nm,'Color','k','FontSize',8)
text(0.55,-0.15+0.05,'Kgm','Color','k','FontSize',8)

text(0.3,-0.2+0.05,'R =','Color','k','FontSize',8)
text(0.4,-0.2+0.05,nR,'Color','k','FontSize',8)
text(0.55,-0.2+0.05,'m','Color','k','FontSize',8)

text(0.3,-0.25+0.05,'v =','Color','k','FontSize',8)
text(0.4,-0.25+0.05,nv,'Color','k','FontSize',8)
text(0.55,-0.25+0.05,'m/sec','Color','k','FontSize',8)

text(0.3,-0.3+0.05,'ω =','Color','k','FontSize',8)
text(0.4,-0.3+0.05,nom,'Color','k','FontSize',8)
text(0.55,-0.3+0.05,'rad/sec','Color','k','FontSize',8)

text(0.3,-0.35+0.05,'F_k =','Color','k','FontSize',8)
text(0.4,-0.35+0.05,nfk,'Color','k','FontSize',8)
text(0.55,-0.35+0.05,'Kgm/sec^2','Color','k','FontSize',8)

text(0.3+1,-0.15+0.05,'t =','Color','b','FontSize',8)
text(0.4+1,-0.15+0.05,nt,'Color','b','FontSize',8)
text(0.55+1,-0.15+0.05,'sec','Color','b','FontSize',8)

axis([0,3,-0.5,0.5])
axis off

arx1 = [0.85, 0.95]+1.64-0.05;
tell1 = [-0.12, -0.12]-0.03+0.05;

line(arx1, tell1, 'Color','r');

text(0.6+1.9-0.05,-0.1-0.03+0.05,'μv^2','Color','r','FontSize',9)
text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045, 'F_k = (1)','Color','r','FontSize',9)
text(0.62+1.9-0.05, -0.14-0.03+0.05, 'R', 'Color','r','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.06, 'v = ωR (2)','Color','b','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.12, 'F_k = mω^2R (3)','Color','r','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.18, 'F_k = mωv (4)','Color','r','FontSize',9)
pause(ryte)

end

xin=xc+a*cos(phil);
t2=(3.0-xin)/v;

xbar1=xc+[0.005,0.005,-0.005,-0.005];
ybar1=yc+[0.0,-0.075,-0.075,0.0];

for time=0.0:0.01:1.2*t2

    if (stam==1)
        ccl=stam;
        while (ccl==1)
            ccl=stam;
            pause(0.01);
            if (status==1)
                return
            end
        end
    end
end

nt=num2str(0.01*round(100*time));

xbody=xc+a*cos(phil)+Rb*cos(th1)+v*time;
ybody=yc+b*sin(phil)+(Rb-0.6*Rb)*sin(th1);

```

```

xbody1=xc+a*cos(phi1)+0.97*Rb*cos(th1)+v*time;
ybody1=yc+b*sin(phi1)+0.97*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody2=xc+a*cos(phi1)+0.93*Rb*cos(th1)+v*time;
ybody2=yc+b*sin(phi1)+0.93*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody3=xc+a*cos(phi1)+0.85*Rb*cos(th1)+v*time;
ybody3=yc+b*sin(phi1)+0.85*(Rb-0.6*Rb)*sin(th1)+0.002;

xbody4=xc+a*cos(phi1)+0.75*Rb*cos(th1)+v*time;
ybody4=yc+b*sin(phi1)+0.75*(Rb-0.6*Rb)*sin(th1)+0.0025;

xbody5=xc+a*cos(phi1)+0.62*Rb*cos(th1)+v*time;
ybody5=yc+b*sin(phi1)+0.62*(Rb-0.6*Rb)*sin(th1)+0.0025;

xbody6=xc+a*cos(phi1)+0.48*Rb*cos(th1)+v*time;
ybody6=yc+b*sin(phi1)+0.48*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody7=xc+a*cos(phi1)+0.32*Rb*cos(th1)+v*time;
ybody7=yc+b*sin(phi1)+0.32*(Rb-0.6*Rb)*sin(th1)+0.003;

xbody8=xc+a*cos(phi1)+0.14*Rb*cos(th1)+v*time;
ybody8=yc+b*sin(phi1)+0.14*(Rb-0.6*Rb)*sin(th1)+0.0035;

xbody9=xc+a*cos(phi1)+0.02*Rb*cos(th1)+v*time;
ybody9=yc+b*sin(phi1)+0.02*(Rb-0.6*Rb)*sin(th1)+0.0035;

xk=xc+a*cos(phi1)+v*time;
yk=yc+b*sin(phi1);

xbar2=xk+[0.005,0.005,-0.005,-0.005];
ybar2=yk+[0.0,-0.095,-0.095,0.0];

xvectorv=xk+[0.004*cos(phi1),0.004*cos(phi1)-Lvx*sin(phi1)/ab,...
0.004*cos(phi1)-Lvx*sin(phi1)/ab+0.04*a*cos(phi1)/ab,...
-(Lvx+0.08*a)*sin(phi1)/ab,...
-0.004*cos(phi1)-Lvx*sin(phi1)/ab-0.04*a*cos(phi1)/ab,...
-0.004*cos(phi1)-Lvx*sin(phi1)/ab,-0.004*cos(phi1)];
yvectorv=yk+[0.004*sin(phi1),0.004*sin(phi1)+Lvy*cos(phi1)/ab,...
0.004*sin(phi1)+Lvy*cos(phi1)/ab+0.04*b*sin(phi1)/ab,...
(Lvy+0.08*b)*cos(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab-0.04*b*sin(phi1)/ab,...
-0.004*sin(phi1)+Lvy*cos(phi1)/ab,-0.004*sin(phi1)];

xgv=xc+1.2*a*cos(phi1+pi/10)+v*time;
ygv=yc+1.2*b*sin(phi1+pi/10);

fill(xtoixos,ytoixos,[0.6078 0.7882 0.9294],...
xpivakasb,ypivakasb,[0.5,0.4,0.3],...
xpivakas,ypivakas,[0.95,0.95,0.95],...
xstick,ystick,[0.5,0.5,0.5],...
xtroxia,ytroxia,[0.2,0.2,0.2],...
xbar1,ybar1,[0.2,0.2,0.2],...
xvectorv,yvectorv,[0.0,0.0,1.0],...
xbody,ybody,[0.7,0.7,0.2],...
xbody,ybody,[0.4,0.4,0.],...
xbody1,ybody1,[0.42,0.42,0.2],...
xbody2,ybody2,[0.44,0.44,0.2],...
xbody3,ybody3,[0.46,0.46,0.2],...
xbody4,ybody4,[0.49,0.49,0.2],...
xbody5,ybody5,[0.52,0.52,0.2],...
xbody6,ybody6,[0.56,0.56,0.2],...
xbody7,ybody7,[0.6,0.6,0.2],...
xbody8,ybody8,[0.65,0.65,0.2],...
xbody9,ybody9,[0.7,0.7,0.2],...
'Linestyle','None')

hold on

fill(xxeri+Dx,yxeri+Dy,[0.45,0.45,0.1],...
xxeri6+Dx,yxeri6+Dy,[0.45,0.45,0.1],...
xxeri1+Dx,yxeri1+Dy,[0.45,0.45,0.1],...
xxeri2+Dx,yxeri2+Dy,[0.45,0.45,0.1],...
xxeri3+Dx,yxeri3+Dy,[0.45,0.45,0.1],...
xxeri4+Dx,yxeri4+Dy,[0.45,0.45,0.1],...
xxeri5+Dx,yxeri5+Dy,[0.45,0.45,0.1])

hold on

```

```

fill(xtroxia,ytroxia,[0.2,0.2,0.2],...
     xbar2,ybar2,[0.2,0.2,0.2],...
     xvectorv,yvectorv,[1.0,1.0,0.0],...
     xbody,ybody,[0.7,0.7,0.2],...
     xbody,ybody,[0.4,0.4,0.],...
     xbody1,ybody1,[0.42,0.42,0.2],...
     xbody2,ybody2,[0.44,0.44,0.2],...
     xbody3,ybody3,[0.46,0.46,0.2],...
     xbody4,ybody4,[0.49,0.49,0.2],...
     xbody5,ybody5,[0.52,0.52,0.2],...
     xbody6,ybody6,[0.56,0.56,0.2],...
     xbody7,ybody7,[0.6,0.6,0.2],...
     xbody8,ybody8,[0.65,0.65,0.2],...
     xbody9,ybody9,[0.7,0.7,0.2],...
     'LineStyle','None')

hold off
% text(0.8,-0.07,'ΟΜΑΔΗ ΚΥΚΛΙΚΗ ΚΙΝΗΣΗ - Κεντρομόλος Δύναμη ','Color','b','FontSize',9)
% text(1.75,-0.07,'Γραμμική Ταχύτητα','Color','k','FontSize',8)

text(xgv,ygv,'v','Color','y','FontSize',8)

text(0.3,-0.15+0.05,'m =','Color','k','FontSize',8)
text(0.4,-0.15+0.05,nm,'Color','k','FontSize',8)
text(0.55,-0.15+0.05,'Kgm','Color','k','FontSize',8)

text(0.3,-0.2+0.05,'R =','Color','k','FontSize',8)
text(0.4,-0.2+0.05,nR,'Color','k','FontSize',8)
text(0.55,-0.2+0.05,'m','Color','k','FontSize',8)

text(0.3,-0.25+0.05,'v =','Color','k','FontSize',8)
text(0.4,-0.25+0.05,nv,'Color','k','FontSize',8)
text(0.55,-0.25+0.05,'m/sec','Color','k','FontSize',8)

text(0.3,-0.3+0.05,'ω =','Color','k','FontSize',8)
text(0.4,-0.3+0.05,nom,'Color','k','FontSize',8)
text(0.55,-0.3+0.05,'rad/sec','Color','k','FontSize',8)

text(0.3,-0.35+0.05,'F_k =','Color','k','FontSize',8)
text(0.4,-0.35+0.05,nfK,'Color','k','FontSize',8)
text(0.55,-0.35+0.05,'Kgm/sec^2','Color','k','FontSize',8)

text(0.3+1,-0.15+0.05,'t =','Color','b','FontSize',8)
text(0.4+1,-0.15+0.05,nt,'Color','b','FontSize',8)
text(0.55+1,-0.15+0.05,'sec','Color','b','FontSize',8)

axis([0,3,-0.5,0.5])
axis off

arx1 = [0.85, 0.95]+1.64-0.05;
tell1 = [-0.12, -0.12]-0.03+0.05;

line(arx1, tell1, 'Color','r');

text(0.6+1.9-0.05,-0.1-0.03+0.05,'mu^2','Color','r','FontSize',9)
text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045, 'F_k = (1)', 'Color','r','FontSize',9)
text(0.62+1.9-0.05, -0.14-0.03+0.05, 'R', 'Color','r','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.06, 'u = ωR (2)', 'Color','b','FontSize',9)

text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.12, 'F_k = mω^2R (3)', 'Color','r','FontSize',9)
text(0.3+1.9-0.03+0.15, -0.12-0.03+0.045-0.18, 'F_k = mωv (4)', 'Color','r','FontSize',9)

% text(1.1,-0.35+0.14,'Η κεντρομόλος δύναμη ενός κινητού είναι ανάλογη της
μάζας','Color','b','FontSize',8)
% text(1.1,-0.38+0.14,'και του τετραγώνου της γραμμικής ταχύτητας όταν η
ακτίνα','Color','b','FontSize',8)
% text(1.1,-0.41+0.14,'της κυκλικής τροχιάς παραμένει σταθερή','Color','b','FontSize',8)

text(1.1-0.2,-0.35+0.01,'Αν η κεντρομόλος δύναμη γίνει μεγαλύτερη από την τάση θραύσης ','Color',[0
0.5 0],'FontSize',8)
text(1.1-0.2,-0.38+0.01,'του νήματος τότε αυτό κόβεται και η σφαίρα κινείται ευθύγραμμα
','Color',[0 0.5 0],'FontSize',8)
text(1.1-0.2,-0.41+0.01,'κατά τη διεύθυνση της γραμμικής ταχύτητας','Color',[0 0.5 0],'FontSize',8)

pause(2*ryte)
end

end

```

```

set(handles.pushbutton3,'enable','on')
set(handles.pushbutton4,'enable','on')
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%
%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%%

% --- Executes on button press in pushbutton2.
function pushbutton2_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global stam;

if (stam==0)
    set(handles.pushbutton2,'string','Συνέχεια')
    set(handles.pushbutton3,'enable','on')
    set(handles.pushbutton4,'enable','on')
    stam=1;
elseif (stam==1)
    set(handles.pushbutton2,'string','Διακοπή')
    set(handles.pushbutton3,'enable','off')
    set(handles.pushbutton4,'enable','off')
    stam=0;
else
end
guidata(hObject, handles);

% --- Executes on button press in pushbutton3.
function pushbutton3_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global t;
global m;
global omega;
global v;
global RR;
global ryt;

axes(handles.axes1)
axis off;
cla

global status;
status=1;

clear t;
clear m;
clear omega;
clear v;
clear RR;
clear ryt;

set(handles.edit1,'enable','on','string','3');
set(handles.edit2,'enable','on','string','');
set(handles.edit3,'enable','on','string','');
set(handles.edit4,'enable','on','string','');
set(handles.edit5,'enable','on','string','10');
set(handles.edit6,'enable','on','string','0.5');

set(handles.pushbutton1,'enable','on');
guidata(hObject, handles);

% --- Executes on button press in pushbutton4.
function pushbutton4_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
global status;
hfin=questdlg('Εξοδος από το πρόγραμμα;');
switch hfin
    case 'Yes'
        status=1;
        closereq;
end

% --- Executes on button press in pushbutton5.
function pushbutton5_Callback(hObject, eventdata, handles)
% hObject    handle to pushbutton5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
! help_Prosoimoiosis.pdf

```

ΒΙΒΛΙΟΓΡΑΦΙΑ

Χρηστός Σαραγιώτης & Αθανάσιος Αρβανιτίδης, *Λογισμός I και Γραμμική Άλγεβρα με MATLAB, Εργαστηριακές Σημειώσεις*, Σέρρες 2006

Δημήτριος Βαρσάμης, *Matlab –Graphical User Interfaces, Σημειώσεις GUI για το εργαστήριο του μαθήματος Ασαφή Συστήματα*, Σέρρες 2006

Ευάγγελος Χατζίκος, *Matlab 7 για μηχανικούς*, Εκδόσεις Τζιόλα

Νίκος Σκουλίδης & Χαρίτων Πολάτογλου, *Δυναμική Μελέτη της Ομαλής Κυκλικής Κίνησης με το Interactive Physics*, 3ο Πανελλήνιο Εκπαιδευτικό Συνέδριο Ημαθίας

<http://el.wikipedia.org/wiki>

<http://ebooks.edu.gr/modules/ebook/show.php>

<http://physiclessons.blogspot.com>