

ΔΥΝΑΜΙΚΟΣ ΠΡΟΓΡΑΜΜΑΤΙΣΜΟΣ

ΑΝΑΦΟΡΑ 1ης ΕΡΓΑΣΤΗΡΙΑΚΗΣ ΑΣΚΗΣΗΣ

Σοφιανόπουλος Έκτορας 2017010016

<https://colab.research.google.com/drive/1F8ZaL-sdYZHtJhJyMjUGmOycmfVACSnS?usp=sharing>

Ως μέρος της άσκησης παραδόθηκε ένα python script μέσω του οποίου υπολογίζεται και απεικονίζεται η συμπεριφορά αυτοκινήτου υπό διαφορετικές συνθήκες σε βάθος χρόνου. Στην αρχή του script δηλώνονται οι παράμετροι του αυτοκινήτου και οι αρχικές καταστάσεις κίνησης. Έπειτα, στη συνάρτηση dynamics, υπολογίζονται οι παράγωγοι της ταχύτητας στον y άξονα καθώς και της γωνιακής ταχύτητας, βάσει της γωνίας στρέψης ($\delta = 0.05 \text{ rad}$) και των πλευρικών δυνάμεων που ασκούνται στους τροχούς, ενώ στη συνάρτηση trajectory_cal υπολογίζεται η τροχιά του οχήματος ως συνάρτηση των μεταβλητών κατάστασης. Ύστερα, σε κάθε επανάληψη του κύριου βρόγχου του κώδικα υπολογίζεται, καταγράφεται σε array και έπειτα ενημερώνεται η παράγωγος της τρέχουσας κατάστασης. Το script τελειώνει οπτικοποιώντας τα arrays που περιέχουν τα αποτελέσματα των παραπάνω υπολογισμών, συγκεκριμένα την τροχιά ανα timestep.

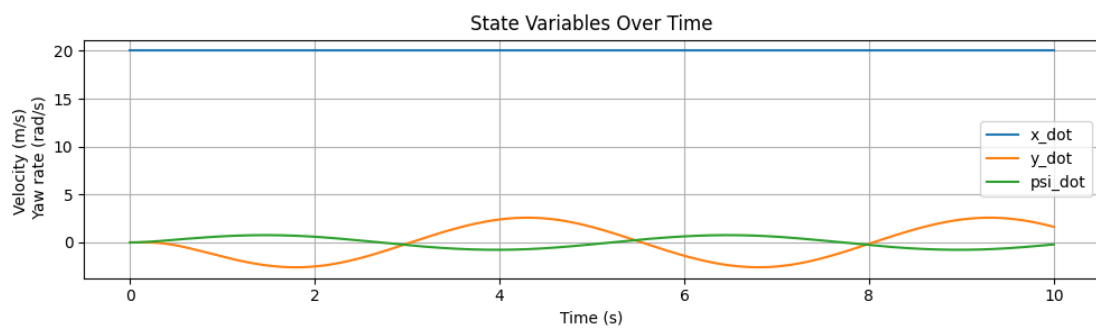
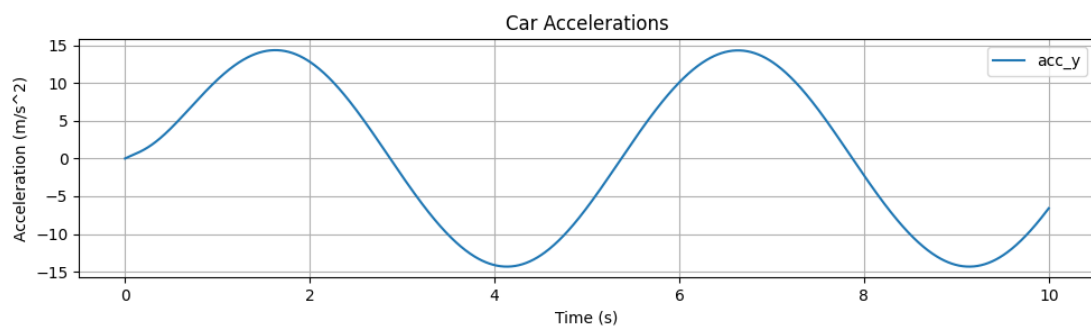
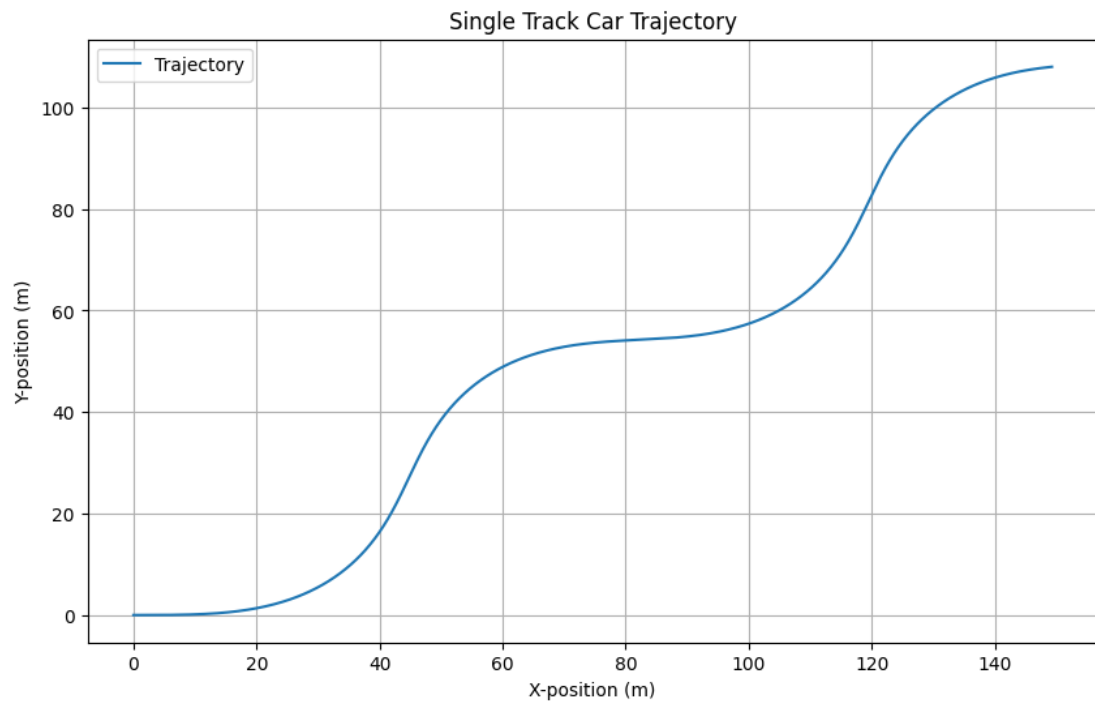
Παρατηρήθηκε πως ενώ έχει δηλωθεί ως σταθερά η V_x velocity στον x άξονα στη συνέχεια του κώδικα αρχικοποιείται ξανά με άλλη τιμή και όνομα x_dot0 . Η διόρθωση που πραγματοποιήθηκε είναι :

$$x_dot0 = V_x$$

Τμήμα Α

Ως πρώτο task έχουμε να προσθέσουμε στο output του script γραφικές παραστάσεις της επιτάχυνσης στους άξονες x, y καθώς και γραφικές των μεταβλητών κατάστασης, ανά timestep του χρόνου της προσομοίωσης. Για να γίνει αυτό, αρκεί να υπολογιστεί και να αποθηκευτεί (σε array) η επιτάχυνση και οι μεταβλητές σε κάθε επανάληψη του main βρόγχου. Η επιτάχυνση στον x άξονα δεν υπολογίζεται (commented out) καθώς η ταχύτητα έχει σταθερή τιμή καθ' όλη τη διάρκεια της προσομοίωσης.

Το δεύτερο task είναι να αλλάξουμε το script έτσι ώστε η γωνία στρέψης του οχήματος να μην έχει πια σταθερή τιμή $\delta = 0.05 \text{ rad}$ αλλά να μεταβάλλεται ως ημίτονο με συχνότητα 0.2 Hz και πλάτος 1.0016 rad . Ο τύπος υπολογισμού είναι $A(\sin(2\pi f i))$, όπου i το κάθε timestep. Επιπλέον χρειάζεται να προσθέσουμε την delta στα arguments της συνάρτησης “dynamics” καθώς πρέπει κάθε φορά που εκτελείται να λαμβάνει υπ' όψη τις αλλαγές του delta.



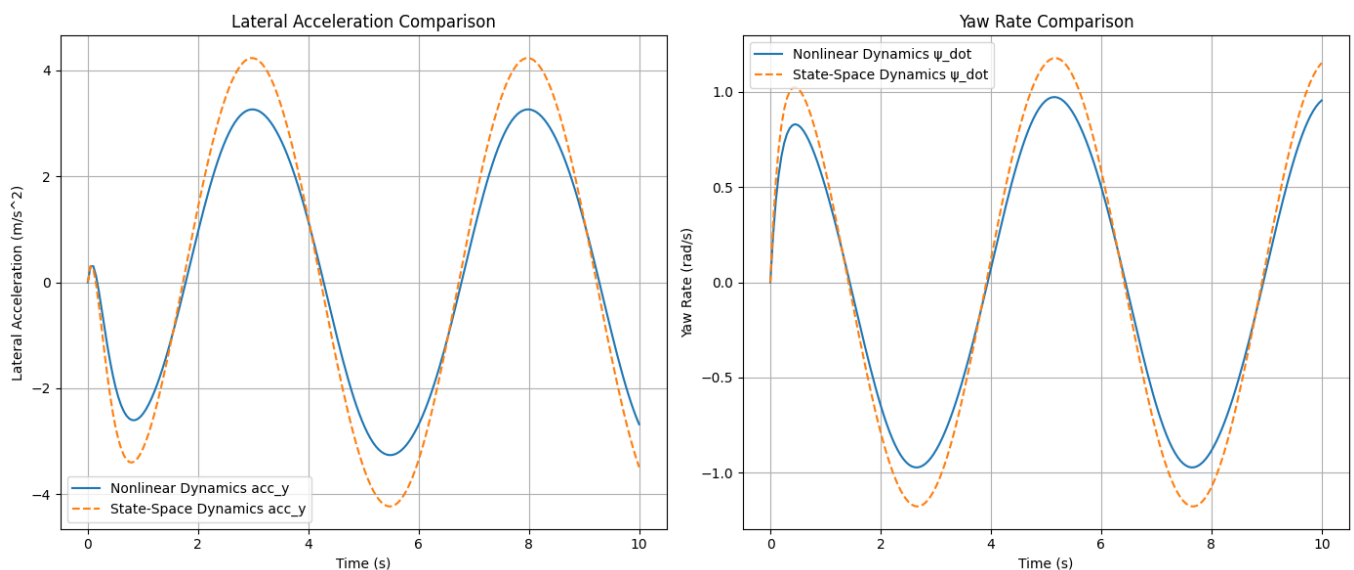
Γραφικές προσομοίωσης Τμήμα Α.

Τμήμα Β

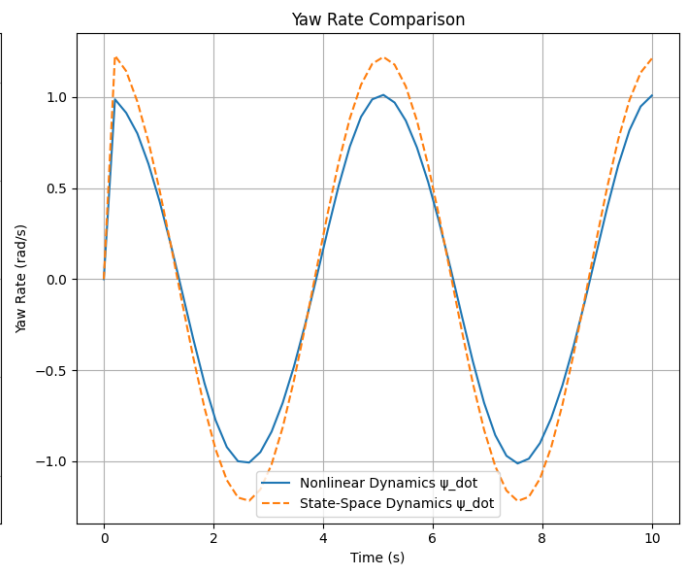
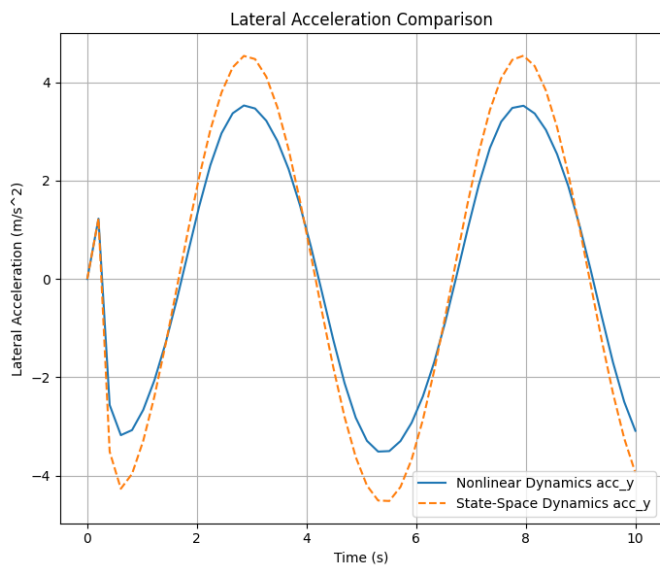
Αρχικά καλούμαστε να μεταβάλλουμε το script έτσι ώστε να παράγει αποτελέσματα χρησιμοποιώντας μη γραμμικό καθώς και γραμμικό μοντέλο υπολογισμών state space :

$$\frac{d}{dt} \begin{Bmatrix} \dot{y} \\ \dot{\psi} \end{Bmatrix} = \underbrace{\begin{bmatrix} -\frac{C_f+C_r}{m\dot{x}} & \frac{C_rl_r-C_fl_f}{m\dot{x}} - \dot{x} \\ \frac{C_rl_r-C_fl_f}{I_z\dot{x}} & \frac{C_fl_f^2+C_rl_r^2}{I_z\dot{x}} \end{bmatrix}}_A \cdot \underbrace{\begin{bmatrix} \dot{y} \\ \dot{\psi} \end{bmatrix}}_B + \begin{bmatrix} C_f/m \\ C_fl_f/I_z \end{bmatrix} \delta$$

Για να επιτευχθεί αυτό προσθέτουμε ακόμη μια συνάρτηση η οποία αναπαράγει την παραπάνω συνάρτηση πινάκων για τις τιμές της προσομοίωσης και με μεταβλητό delta. Η συνάρτηση δέχεται σαν όρισμα έναν μονοδιάστατο πίνακα που περιέχει τις τιμές των y_dot , ψ_dot και επιστρέφει έναν πίνακα με τις παραγώγους των τιμών σε κάθε επανάληψη του κύριου βρόγχου. Έξω από το loop έχουμε αρχικοποιήσει νέες arrays για την αποθήκευση των αποτελεσμάτων. Οι δύο δυναμικές συναρτήσεις δουλεύουν με αντίστοιχο τρόπο και διαφέρουν στον υπολογισμό των παραγώγων καθώς και στο μέγεθος των arguments.



$dt = 0.05 \text{ sim}$



$dt = 0.2 \text{ sim}$

Όπως φανερώνουν τα διαγράμματα, οι δύο μέθοδοι έχουν παρόμοιο output και στις δυο προσομοιώσεις. Παρατηρείται μια διαφορά φάσης και πλάτους μεταξύ των καμπυλών η οποία οφείλεται στην πιο σύνθετη και με μεγαλύτερη ακρίβεια (ευαισθησία στις μεταβολές της γωνίας στρέψης) του μη γραμμικού μοντέλου.