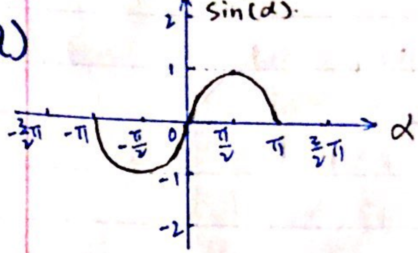
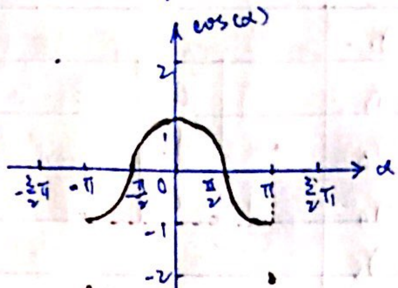
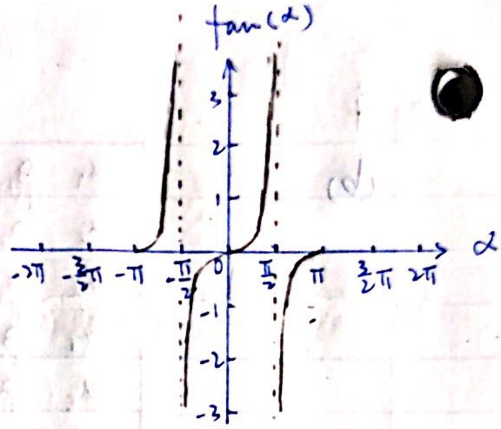
5. Kinematic Model for a Simple Car

(a)







Yes. I can. As we could see, for small values of α, the value of $\sin(\alpha)$ is really close to 0 on the sin graph. Similarly, on the graphs of $\cos(\alpha)$, $\tan(\alpha)$, for small values of α ($\alpha \approx 0$); $\cos(\alpha)$ is really close to 1, and $\tan(\alpha)$ is really close to 0, which justifies the approximations.

(b). Let $A = \begin{bmatrix} a_{11} & a_{12} & a_{13} & a_{14} \\ a_{21} & a_{22} & a_{23} & a_{24} \\ a_{31} & a_{32} & a_{33} & a_{34} \\ a_{41} & a_{42} & a_{43} & a_{44} \end{bmatrix}$ and $B = \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \\ b_{31} & b_{32} \\ b_{41} & b_{42} \end{bmatrix}$ and using the given information, with approximation that $\theta, \varphi \approx 0$. so, $\sin\theta = \tan\varphi \approx 0$, $\cos\theta \approx 1$, we have that:

$$a_{11} \cdot x[k] + a_{12} \cdot y[k] + a_{13} \cdot \theta[k] + a_{14} \cdot v[k] + b_{11} \cdot a[k] + b_{12} \cdot \varphi[k] = x[k+1] = x[k] + v[k]\cos(\theta[k])\Delta t.$$
$$a_{21} \cdot x[k] + a_{22} \cdot y[k] + a_{23} \cdot \theta[k] + a_{24} \cdot v[k] + b_{21} \cdot a[k] + b_{22} \cdot \varphi[k] = y[k+1] = y[k] + v[k]\sin(\theta[k])\Delta t.$$
$$a_{31} \cdot x[k] + a_{32} \cdot y[k] + a_{33} \cdot \theta[k] + a_{34} \cdot v[k] + b_{31} \cdot a[k] + b_{32} \cdot \varphi[k] = \theta[k+1] = \theta[k] + \frac{v[k]}{L}\tan(\varphi[k])\Delta t$$
$$a_{41} \cdot x[k] + a_{42} \cdot y[k] + a_{43} \cdot \theta[k] + a_{44} \cdot v[k] + b_{41} \cdot a[k] + b_{42} \cdot \varphi[k] = v[k+1] = v[k] + a[k]\Delta t.$$

With given information that $L = 1.0$ m and $\Delta t = 0.1$ s, so we have that:

$$x[k+1] = 1 \cdot x[k] + 0 \cdot y[k] + 0 \cdot \theta[k] + 0.1 \cdot v[k] + 0 \cdot a[k] + 0 \cdot \varphi[k].$$
$$y[k+1] = 0 \cdot x[k] + 1 \cdot y[k] + 0 \cdot \theta[k] + 0 \cdot v[k] + 0 \cdot a[k] + 0 \cdot \varphi[k]$$
$$\theta[k+1] = 0 \cdot x[k] + 0 \cdot y[k] + 1 \cdot \theta[k] + 0 \cdot v[k] + 0 \cdot a[k] + 0 \cdot \varphi[k].$$
$$v[k+1] = 0 \cdot x[k] + 0 \cdot y[k] + 0 \cdot \theta[k] + 1 \cdot v[k] + 0.1 \cdot a[k] + 0 \cdot \varphi[k].$$

Thus, $a_{11} = a_{22} = a_{33} = a_{44} = 1$, $a_{14} = 0.1$, $b_{41} = 0.1$

$a_{12} = a_{13} = a_{21} = a_{23} = a_{24} = a_{31} = a_{32} = a_{34} = a_{41} = a_{42} = a_{43} = 0.$

$b_{11} = b_{12} = b_{21} = b_{22} = b_{31} = b_{32} = b_{42} = 0.$

Therefore, $A = \begin{bmatrix} 1 & 0 & 0 & 0.1 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}$ and $B = \begin{bmatrix} 0 & 0 \\ 0 & 0 \\ 0 & 0 \\ 0.1 & 0 \end{bmatrix}$

(c) They are very similar.

Since the steering angle is so small ($\phi[k] = 0.0001$ rad), its tangent value could be approximated as 0, which is what I did in my linear approximation. In other words, because the steering angle is so small, the nonlinear system behaves just like a linear one.

(d) They are very different.

Since the steering angle is pretty big ($\phi[k] = 0.5$ rad), its tangent value could **not** be approximated as 0, which is what I did in my linear approximation. In other words, the nonlinear system in IPython takes the rather large steering angle into consideration, while my linear approximation doesn't, which is why the trajectories differ.