Robotic Interception of Moving Objects Using an Augmented Ideal Proportional Navigation Guidance Technique

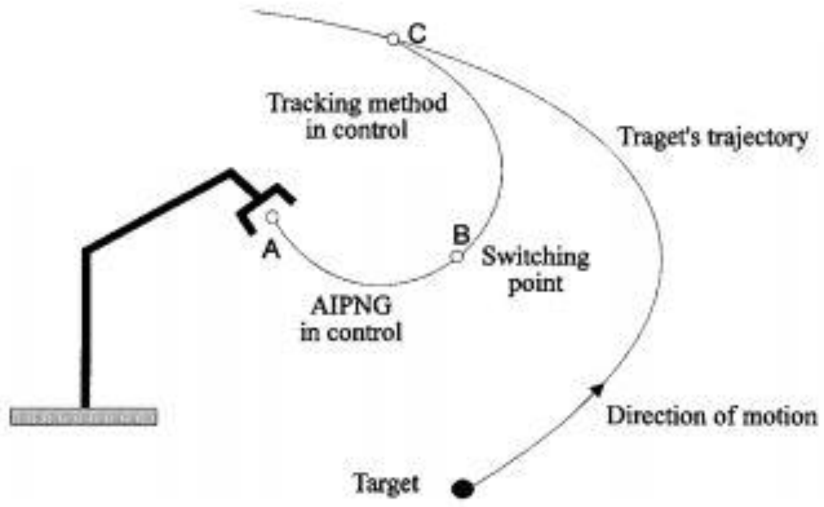
Authored by :

- Mehran Mehrandezh, Member, IEEE
- Naftali M. Sela
- Robert G. Fenton
- Beno Benhabib, Member, IEEE

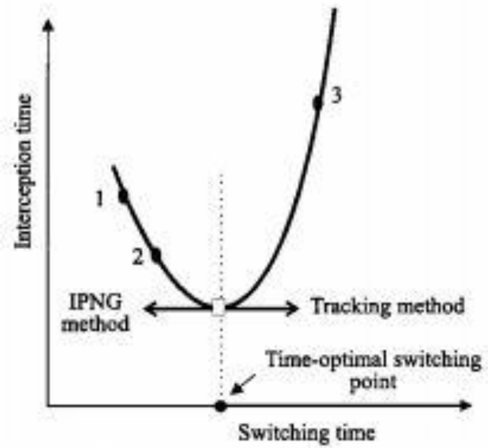
Team Members Intro

1. Nitesh Kumar (ME16B063)
2. Subham Swastik Samal (ME16B045)
3. Pravesh Rana Vadapalli (NA15B042)

Theory of paper



$$a_{IPNG} = \lambda \dot{\mathbf{r}} \times \dot{\theta}_{LOS}$$



In (1), $\dot{\theta}_{\text{LOS}}$ can also be expressed as a function of $\boldsymbol{r}$ and $\dot{\boldsymbol{r}}$ as follows:

$$\dot{\theta}_{\text{LOS}} = \left\{ \frac{\boldsymbol{r} \times \dot{\boldsymbol{r}}}{|\boldsymbol{r}|^2} \right\}. \tag{2}$$

By substituting (2) into (1), one obtains

$$\boldsymbol{a}_{\text{IPNG}} = \frac{\lambda}{|\boldsymbol{r}|^2} \{ \dot{\boldsymbol{r}} \times (\boldsymbol{r} \times \dot{\boldsymbol{r}}) \}. \tag{3}$$

Since $\dot{\boldsymbol{r}} \times (\boldsymbol{r} \times \dot{\boldsymbol{r}}) = \boldsymbol{r}(\dot{\boldsymbol{r}}.\dot{\boldsymbol{r}}) - \dot{\boldsymbol{r}}(\boldsymbol{r}.\dot{\boldsymbol{r}})$, (3) can be rewritten as

$$\boldsymbol{a}_{\text{IPNG}} = K_d(\boldsymbol{r}, \dot{\boldsymbol{r}}, \lambda)\dot{\boldsymbol{r}} + K_p(\boldsymbol{r}, \dot{\boldsymbol{r}}, \lambda)\boldsymbol{r} \tag{4}$$

where K_d and K_p are calculated as

$$K_p(\boldsymbol{r}, \dot{\boldsymbol{r}}, \lambda) = \lambda \left(\frac{|\dot{\boldsymbol{r}}|}{|\boldsymbol{r}|} \right)^2, \quad K_d(\boldsymbol{r}, \dot{\boldsymbol{r}}, \lambda) = -\lambda \left(\frac{(\boldsymbol{r}.\dot{\boldsymbol{r}})}{|\boldsymbol{r}|^2} \right). \tag{5}$$

$$\boldsymbol{a}_c = \boldsymbol{a}_{\text{IPNG}} + \beta \boldsymbol{U}_{\text{LOS}} \tag{6}$$

where U_{LOS} is the unit vector in the LOS direction and β is a scalar whose value is computed according to

$$\beta = \max \left(\bigcap_{i=1}^{n} H_i \right), \quad H_i = \{ \beta \| T_i | \leq \alpha | T_{i\,\max} | \}, \quad i = 1, 2, \cdots, n. \tag{7}$$

$$\boldsymbol{a}_{\text{AIPNG}} = \boldsymbol{a}_{\text{IPNG}} + \boldsymbol{a}_T \equiv K_d \dot{\boldsymbol{r}} + K_p \boldsymbol{r} + \boldsymbol{a}_T$$

$$K_p\boldsymbol{r} + K_d\dot{\boldsymbol{r}} + (\boldsymbol{a}_T - \boldsymbol{a}_{\mathrm{AIPNG}}) = 0$$

and substituting $(\boldsymbol{a}_T - \boldsymbol{a}_{\mathrm{AIPNG}})$ with $\ddot{\boldsymbol{r}}$:

$$\ddot{\boldsymbol{r}} + K_d\dot{\boldsymbol{r}} + K_p\boldsymbol{r} = 0.$$

$$\boldsymbol{a}_c = K_d\dot{\boldsymbol{r}} + K_p\boldsymbol{r} + \boldsymbol{a}_T + \beta(t)\boldsymbol{U}_{\mathrm{LOS}}.$$

$$\boldsymbol{a}_c = K_d\dot{\boldsymbol{r}} + K_p\boldsymbol{r} + \boldsymbol{a}_T + \beta(t)\boldsymbol{U}_{\mathrm{LOS}}. \tag{21}$$

By replacing $(\boldsymbol{a}_T - \boldsymbol{a}_c)$ by $\ddot{\boldsymbol{r}}$ and U_{LOS} by $\boldsymbol{r}/|\boldsymbol{r}|$ and rearranging the remaining terms in (21), one obtains

$$\ddot{\boldsymbol{r}} + K_d\dot{\boldsymbol{r}} + \left(K_p + \frac{\beta(t)}{|\boldsymbol{r}|}\right)\boldsymbol{r} = 0. \tag{22}$$

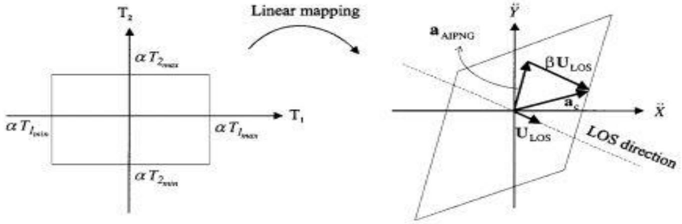


Fig. 3. Upgrading the acceleration command of the AIPNG.

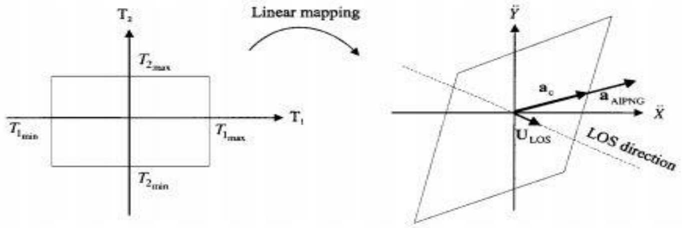
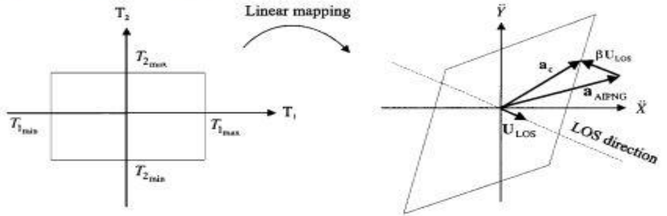
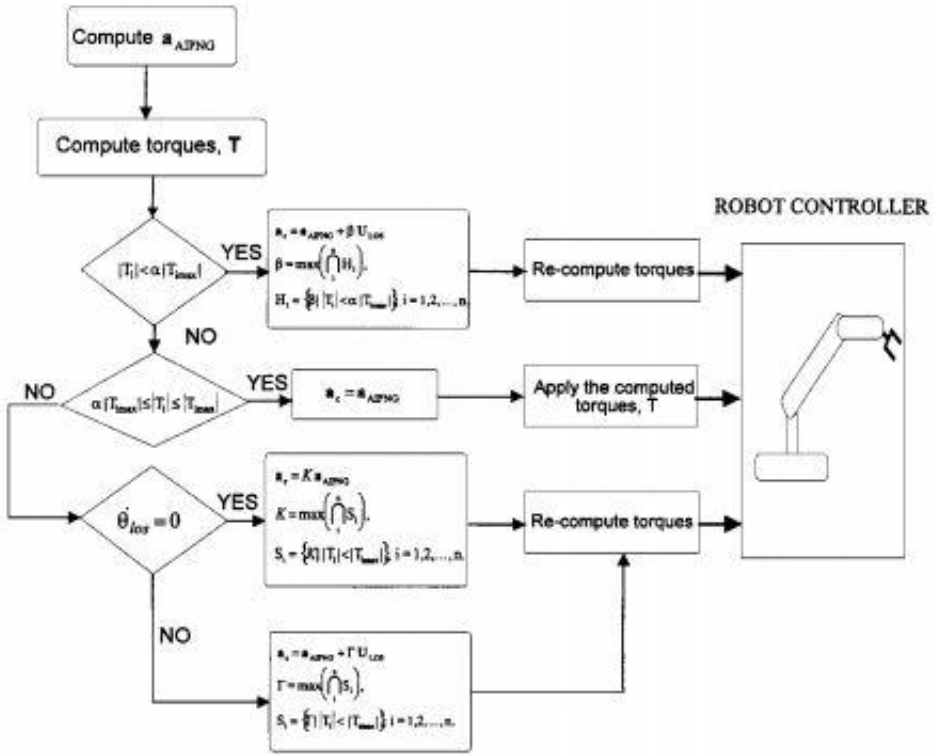
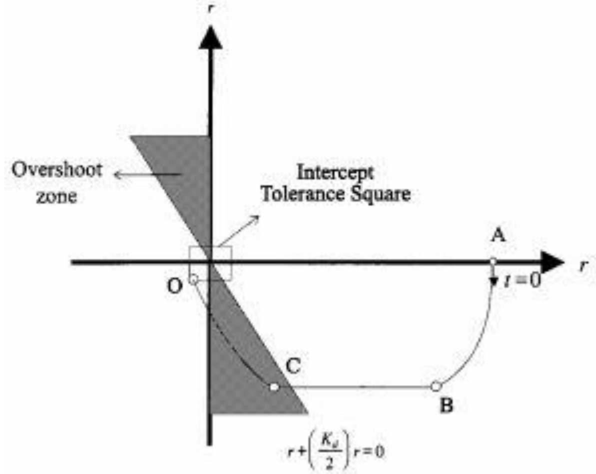


Fig. 4. Limiting the acceleration command of the AIPNG.





Compute a AIPNG
Compute torques, T
YES
NO
YES
NO
YES
NO
Re-compute torques
Apply the computed torques, T
Re-compute torques
ROBOT CONTROLLER

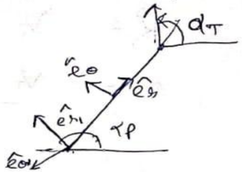


Overshoot zone
Intercept Tolerance Square
A
O
C
B

$$\vec{r} = r\hat{e}_r$$

$$\dot{\vec{r}} = \dot{r}\hat{e}_r + r\dot{\theta}\hat{e}_\theta$$

$$\dot{\vec{r}} = V_R\hat{e}_r + V_\theta\hat{e}_\theta$$



$$a_{IPNG} = \lambda(V_R\hat{e}_r + V_\theta\hat{e}_\theta)\times\dot{\theta}\hat{e}_z$$

$$= -\frac{\lambda V_R V_\theta}{R}\hat{e}_\theta + \lambda\frac{V_\theta^2}{R}\hat{e}_r$$

$$\vec{V_p} = V_p\vec{e}_{r_1}$$

$$\dot{\vec{V_p}} = \dot{V}_p\vec{e}_{r_1} + V_p\dot{\alpha}_p\vec{e}_{\theta_1}$$

Rotated by $-(\alpha_p-\theta)$ angle.

$$a_{IPNG} = \begin{bmatrix} \cos(\alpha_p-\theta) & \sin(\alpha_p-\theta) \\ -\sin(\alpha_p-\theta) & \cos(\alpha_p-\theta) \end{bmatrix} \begin{bmatrix} \frac{\lambda V_\theta^2}{R} \\ \frac{-\lambda V_R V_\theta}{R} \end{bmatrix}$$

$$a_{IPNG} = \left[\frac{\lambda V_\theta^2}{R}\cos(\alpha_p-\theta) - \lambda\frac{V_R V_\theta}{R}\sin(\alpha_p-\theta)\right]\hat{i}$$

$$\left[-\frac{\lambda V_\theta^2}{R}\sin(\alpha_p-\theta) - \frac{\lambda V_R V_\theta}{R}\cos(\alpha_p-\theta)\right]\hat{j}$$

$$\dot{R} = V_T\cos(\alpha_T-\theta) - V_p\cos(\alpha_p-\theta)$$

$$\dot{\theta} = \frac{V_T\sin(\alpha_T-\theta) - V_p\sin(\alpha_p-\theta)}{R}$$

$$\dot{V}_p = \frac{\lambda V_\theta^2}{R}\cos(\alpha_p-\theta) - \frac{\lambda V_R V_\theta}{R}\sin(\alpha_p-\theta)$$

$$\dot{\alpha}_p = \frac{-\lambda V_\theta^2}{R}\sin(\alpha_p-\theta) \; \frac{-\lambda V_R V_\theta}{R}\cos(\alpha_p-\theta)$$

a_{AIPNG} :

$$= \lambda (V_r \hat{e}_r + V_\theta \hat{e}_\theta) \times \dot{\theta} \hat{e}_z + a_{Ti} \hat{i} + a_{Tj} \hat{j}$$

$$= \frac{-\lambda V_R V_\theta}{R} \hat{e}_\theta + \frac{\lambda V_\theta^2}{R} \hat{e}_r + a_{Ti} \hat{i} + a_{Tj} \hat{j}$$

$$= \begin{bmatrix} \cos(\alpha_p - \theta) & \sin(\alpha_p - \theta) \\ -\sin(\alpha_p - \theta) & \cos(\alpha_p - \theta) \end{bmatrix} \begin{bmatrix} \frac{\lambda V_\theta^2}{R} \\ \frac{-\lambda V_R V_\theta}{R} \end{bmatrix} + \begin{bmatrix} \cos\alpha_p & \sin\alpha_p \\ -\sin\alpha_p & \cos\alpha_p \end{bmatrix} \begin{bmatrix} a_{Ti} \\ a_{Tj} \end{bmatrix}$$

$$\dot{V}_p = \frac{\lambda V_\theta^2}{R} \cos(\alpha_p - \theta) - \frac{V_R \lambda V_\theta}{R} \sin(\alpha_p - \theta) + a_{Ti} \cos\alpha_p + a_{Tj} \sin\alpha_p$$

$$\dot{\alpha}_p = -\frac{\lambda V_\theta^2}{R V_p} \sin(\alpha_p - \theta) + \left(-\frac{\lambda V_R V_\theta}{R}\right) \cos(\alpha_p - \theta) - \sin(\alpha_p)\, a_{Ti} + a_{Tj} \cos(\alpha_p)$$

$$V_R = V_T \cos(\alpha_T - \theta) - V_p \cos(\alpha_p - \theta)$$

$$V_\theta = V_T \sin(\alpha_T - \theta) - V_p \sin(\alpha_p - \theta)$$

$$\dot{V}_p = \frac{\lambda V_\theta^2}{R} \cos(\alpha_p - \theta) - \frac{\lambda V_\theta V_R}{R} \sin(\alpha_p - \theta)$$

$$a_{Ti} \cos\alpha_p + a_{Tj} \sin\alpha_p$$

$$\dot{\alpha}_p = \frac{-\lambda V_\theta^2}{R V_p} \sin(\alpha_p - \theta) - \frac{\lambda V_R V_\theta}{R V_p} \cos(\alpha_p - \theta)$$

$$- a_{Tj} \sin\alpha_p + a_{Tj} \cos(\alpha_p)$$

$$\dot{V}_T = \mathrm{Sqrt}(a_{Ti}^2 + a_{Tj}^2) \times \cos\left(\tan^{-1}\left(\frac{a_{Tj}}{a_{Ti}}\right) - \theta\right)$$

$$\dot{\alpha}_T = \mathrm{Sqrt}(a_{Tj}^2 + a_{Tj}^2) \times \sin\left(\tan^{-1}\left(\frac{a_{Tj}}{a_{Ti}}\right) - \theta\right)$$

Simulation Parameters

Initial Vt=0.2236m/s

alphat=-pi/12

theta0=atan(2);

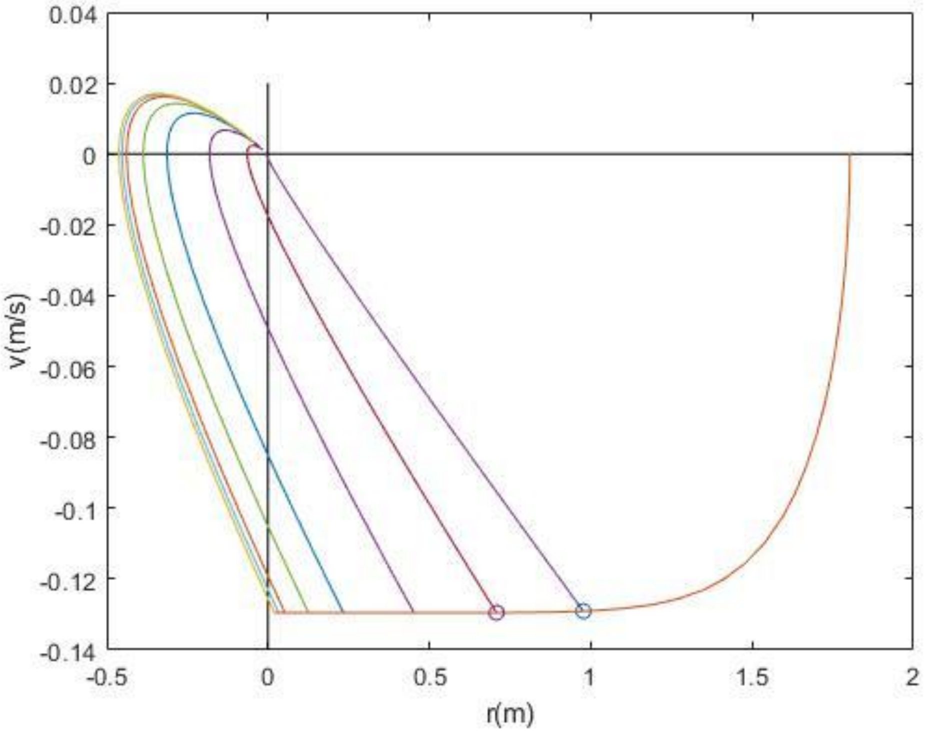
R0=1.8028m

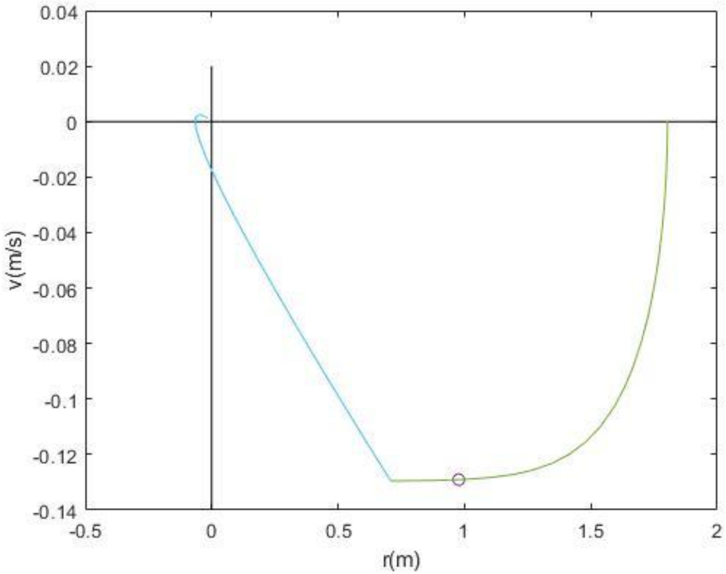
lambda=5

Kp=0.2

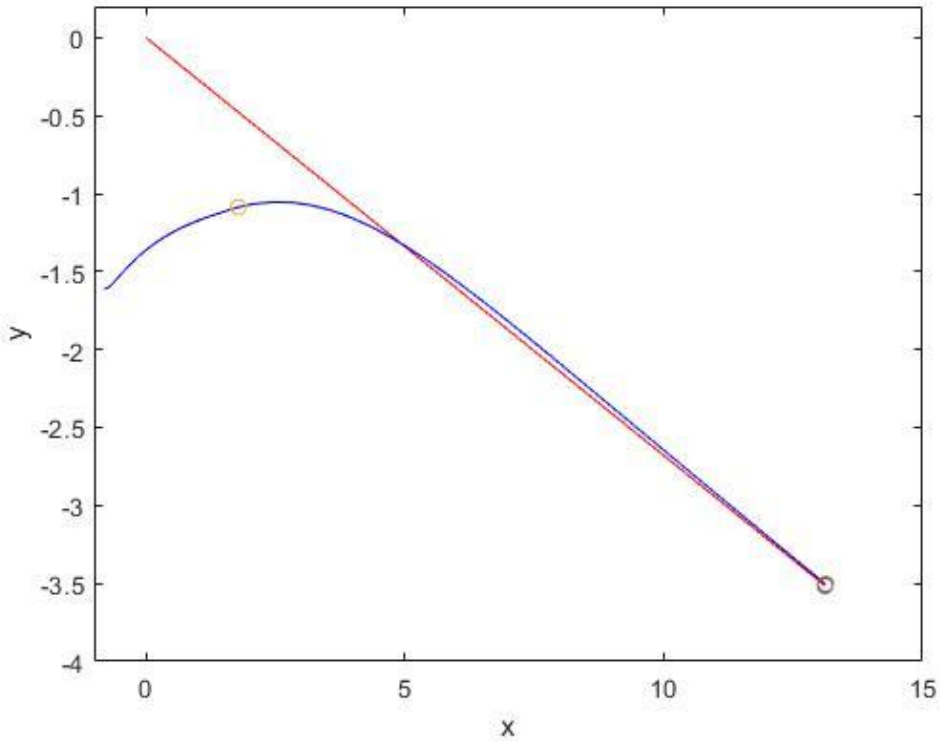
Kd=0.01

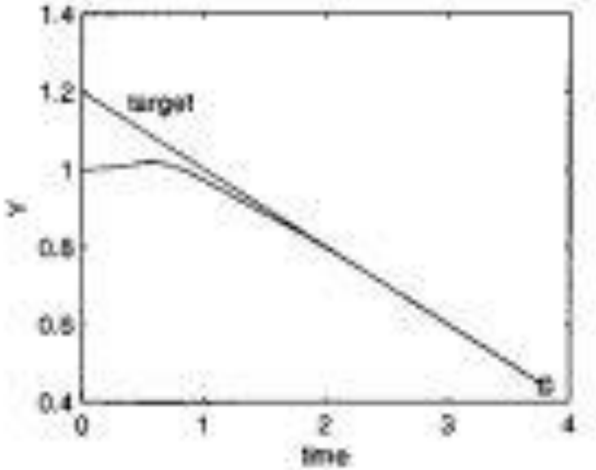
Finding OSP


0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
v(m/s)
-0.5
0
0.5
1
1.5
2
r(m)


0.04
0.02
0
-0.02
-0.04
-0.06
-0.08
-0.1
-0.12
-0.14
v(m/s)
-0.5
0
0.5
1
1.5
2
r(m)

Trajectory





From Paper

AIPNG + Modified CT

Typical CT takes over at point C whereas CT takes over at point E in modified CT method, A - B - C is the same for both methods.

- Seg C - D : Point C represents OSP, $\dot{\theta}_{LOS}$ must have approached zero. Segment C - D represents zero closing acc phase but still moves according to AIPNG.
- Seg D - E : Robot moves with const deceleration in this phase.
- Seg E - O : Robot switches over to PD type CT control. Point E is user defined.

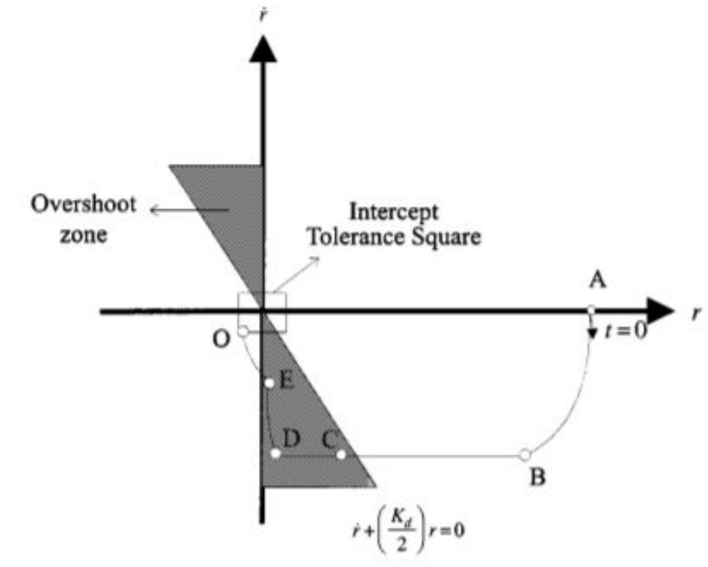


Fig. 10. Phase-plane trajectory of the AIPNG + modified_CT method.

Selecting Point E :

$$r_E = r_o + \gamma(r_c - r_o)$$

$$\frac{r\sqrt{K_p}}{(\dot{r} + \sqrt{K_p})} = \ln\left(\frac{C}{\dot{r} + r\sqrt{K_p}}\right).$$

$$C = \left(\dot{r}_0 + r_0\sqrt{K_p}\right)e^{\left(\frac{r_0\sqrt{K_p}}{\dot{r}_0 + \sqrt{K_p}}\right)},$$

- $\gamma \in [0, 1]$ is user defined. Smaller γ implies point E is closer to point O.
- Smaller γ is preferred as it reduces the interception time.

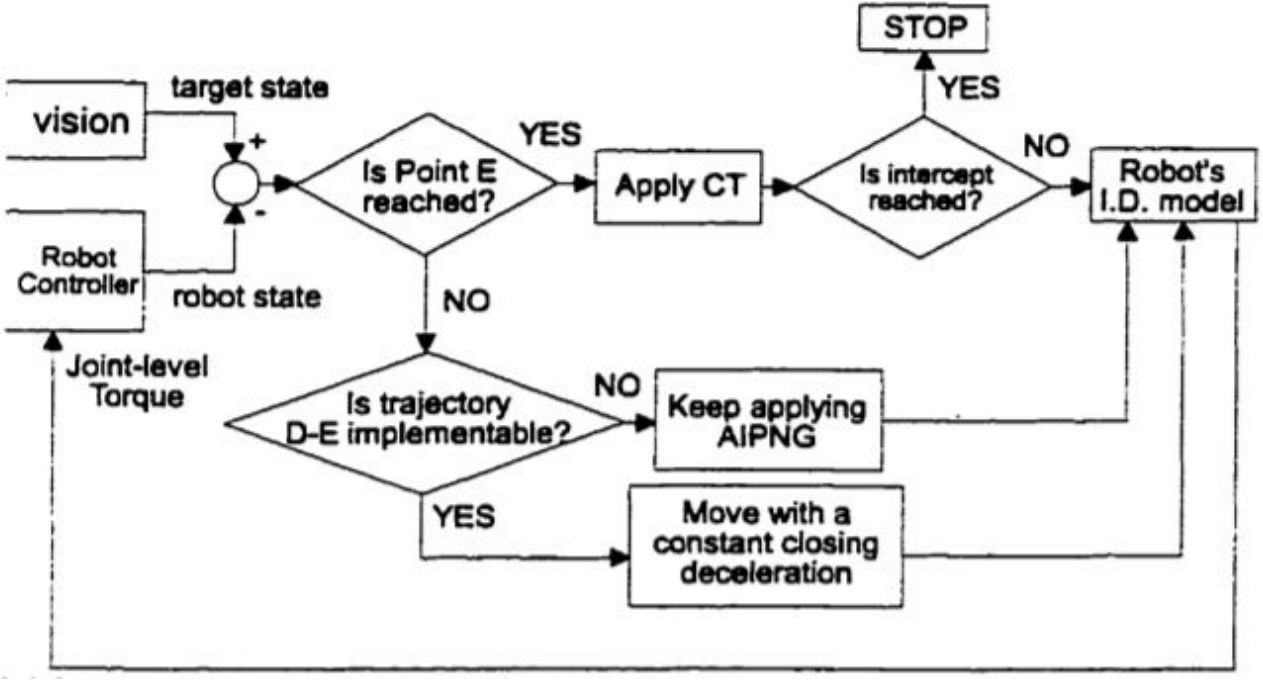
Implementation of Seg D - E

$$\ddot{r}_{constant} = \frac{(\dot{r}_E)^2 - (\dot{r})^2_{t=t_{AIPNG}+i\Delta t}}{2[(r_E) - (r)_{t=t_{AIPNG}+i\Delta t}]}, \quad i = 1,2,\ldots,n,$$

$$\ddot{r}_{permissible} = \frac{\sum_{i=1}^{i} (\ddot{r}_{max})_i + \ddot{r}_E}{i+1},$$

$$\ddot{r}_{constant} \geq \ddot{r}_{permissible}$$

Overall Implementation



Step 0: Is OSP reached? If yes, solve for the Trajectory (C-O). Assign a value to r_E along trajectory C-O, Figure 4.14. Compute the value of $\dot{r}_E$ using Equation (4.46) and go to Step 1. Otherwise, let the robot move as instructed by AIPNG.

Step 1: Set i = 1.

Step 2: Compute the constant deceleration of the robot to bring it from its current state to the state found in Step 0, namely Point E, using Equation (4.48).

Step 3: Compute the permissible deceleration of the robot in the LOS direction using Equation (4.49).

Step 4: Compare the $\ddot{r}_{constant}$, computed in Step 2, with $\ddot{r}_{permissible}$, found in Step 3. If Equation (4.50) is satisfied go to Step 5, otherwise, go to Step 6.

Step 5: Move the robot with $\ddot{r} = 0$ for the next time-step. Set i=i+1. Go to Step 2.

Step 6: Move the robot with $\ddot{r} = \ddot{r}_{constant}$ for the next time-step. Set i=i+1.

Step 7: If $|\dot{r}_i - \dot{r}_E| \le \{(Tol)_v\}_{CT}$ *and* $|r_i - r_E| \le \{(Tol)_p\}_{CT}$, go to Step 8. Otherwise, go to Step 6.

Step 8: Move the robot with $\ddot{r} = -K_d \dot{r}_i - K_p r_i$. If $r \le Tol_p$ *and* $\dot{r} \le Tol_v$, stop the interception scheme. Otherwise, go to Step 9.

Step 9: Set i=i+1. Go to Step 8.