

# **Relazione TORCS**

*Team: I TISCHITOSCHI*

## Introduzione

Il progetto “Torcs” ci è stato utile per applicare, in modo pratico, ciò che abbiamo studiato durante il corso di robotica mobile.

Tramite l'utilizzo di programmi quali MatLab e Simulink abbiamo progettato un controllore **path following** sulla pista “Inferno 1” col veicolo “car1-ow1” che abbiamo schematizzato come un unicycle.

Il modello iniziale da cui siamo partiti era un esempio di Adaptive Cruise Control fornito da GitHub su cui è implementato “TorcsLink”, un interfaccia che permette il collegamento tra Torcs e MatLab/Simulink.

Sulla base di questo esempio abbiamo iniziato ad elaborare la legge di controllo dello sterzo, dell'accelerazione e del cambio.

L'interfaccia “TorcsLink” ci ha fornito in ambiente MatLab e Simulink i dati restituiti da Torcs come bus di dati a partire dai quali abbiamo determinato la legge di controllo sullo sterzo mentre con delle righe di codice in linguaggio C, implementate in una MatLab Function, abbiamo regolato il cambio del veicolo.

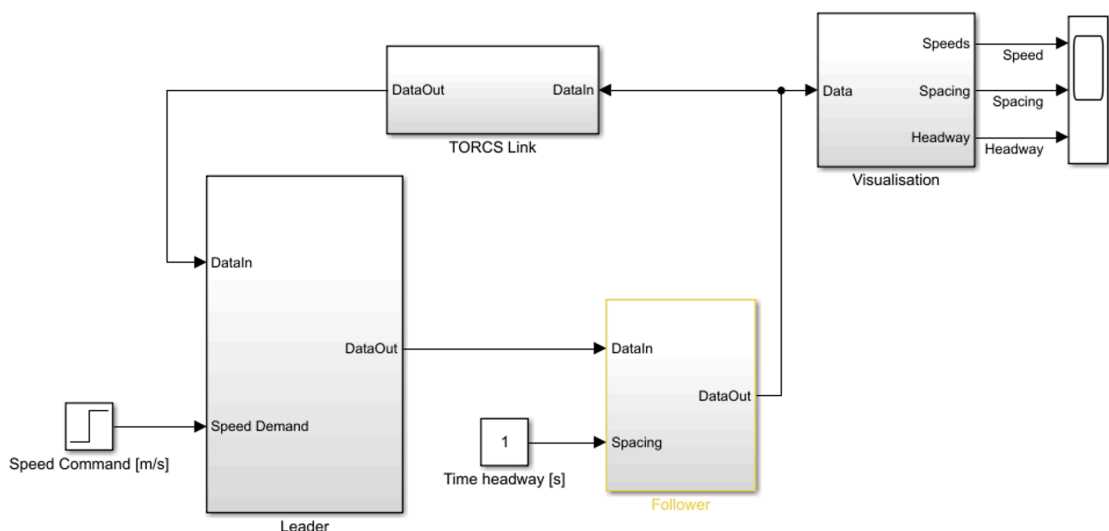
---

### Simple Automatic Cruise Control (ACC) Example

This model simulates a basic Constant Time Headway (CTH) ACC system using The Open Source Race Car Simulink (TORCS) to provide the dynamic model

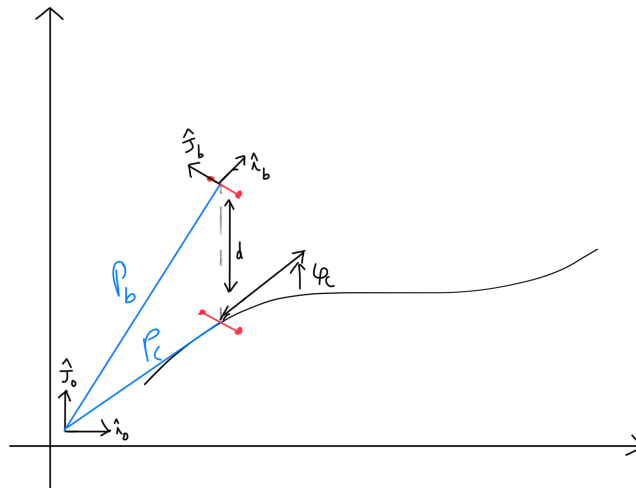
Before running this model, start a TORCS race with the 'matlab 0' and 'matlab 1' in

Both vehicle initially accelerate to 14m/s (50kph), maintaining a 1s headway.  
After 30s the leader accelerates to 25m/s (90kph)  
the follower will increase its spacing to maintain the desired headway



## ***Legge di controllo sullo sterzo***

La legge di controllo sullo sterzo è un **path following** che allinea la posizione dell'uniciclo alla traiettoria della curva regolando lo sterzo  $\varpi$ ; abbiamo determinato la traiettoria della curva nel modo più ottimale possibile, per ridurre i tempi del giro, basandoci sui valori di: "*lateral error*" (distanza dell'uniciclo dal centro della pista), "*heading error*" (angolo tra la verticale dell'uniciclo e la traiettoria), "*road curvature*" (curvatura) e "*road distance*" (ascissa curvilinea).



Fissati i sistemi di riferimento dell'uniciclo e quello di Frenet-Serret

$$F_0 = \{P_0 \quad \hat{i}_0 \quad \hat{j}_0\} = \left\{ \begin{pmatrix} 0 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 1 \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 1 \end{pmatrix} \right\}$$

$$F_c = \{P_c \quad \hat{i}_c \quad \hat{j}_c \quad \hat{k}_c\} = \left\{ \begin{pmatrix} x_c \\ y_c \\ 0 \end{pmatrix} \quad \begin{pmatrix} \cos\varphi_c \\ \sin\varphi_c \\ 0 \end{pmatrix} \quad \begin{pmatrix} -\sin\varphi_c \\ \cos\varphi_c \\ 0 \end{pmatrix} \quad \begin{pmatrix} 0 \\ 0 \\ 1 \end{pmatrix} \right\}$$

siamo pervenuti al modello cinematico dell'uniciclo

$$\begin{bmatrix} \dot{\sigma} \\ \dot{d} \\ \dot{g} \end{bmatrix} = \begin{bmatrix} \frac{\cos\vartheta}{1-d\gamma(\sigma)} & 0 \\ \sin\vartheta & 0 \\ -\frac{\cos\vartheta \cdot \gamma(\sigma)}{1-d\gamma(\sigma)} & 1 \end{bmatrix} \cdot \begin{bmatrix} v \\ \omega \end{bmatrix}$$

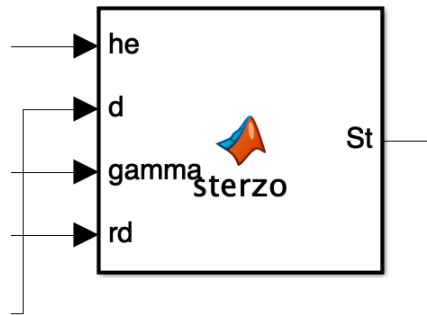
Affinchè l'uniciclo si allinei alla curva abbiamo utilizzato una candidata di Lyapunov quadratica in modo tale che si annulli la distanza dell'uniciclo dalla curva ( $d$ ) e l'angolo  $\vartheta$ , ossia l'angolo ottenuto dalla differenza dell'orientamento del veicolo da quello desiderato:

$$V = \frac{1}{2}d^2 + \frac{1}{2}\vartheta^2$$

da cui abbiamo ottenuto la legge di controllo:

$$\omega = -k \cdot \vartheta - d \cdot v \cdot \text{sinc}(\vartheta) + \frac{\cos \vartheta \cdot \gamma(\sigma)}{1 - d\gamma(\sigma)} \cdot v$$

applicando il Teorema di Krasowskii - Lasalle è risultata essere asintoticamente stabile per il task.



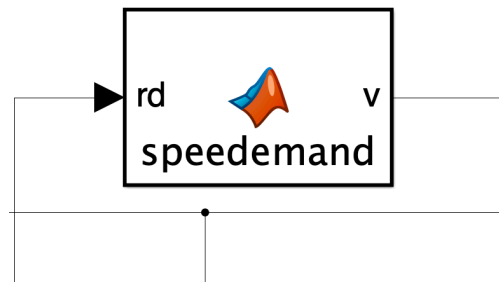
```

1. function St = sterzo(he, d, gamma,rd)
2.
3. k=2.8;
4. he=-he;
5.
6. St= (-k*he- d*sinc(he) + (cos(he)*gamma/(1-d*gamma))) *0.5;
   %legge di controllo
7.
8. if St < -0.785398
9.     St=-0.785398;
10. elseif St > 0.785398
11.     St=0.785398;
12. end
13. St=St/0.785398;
14.
15. if rd>2635 && rd<2820
16.     if St < -0.9
17.         St=-0.9;
18.     elseif St > 0.9
19.         St=0.9;
20.     end
21.
22. else
23.     if St < -0.6
24.         St=-0.6;
25.     elseif St > 0.6
26.         St=0.6;
27.     end
28.
29. end

```

## Controllo della velocità

Per effettuare le curve nel migliore dei modi abbiamo adeguato la velocità di approccio alla curva da parte dell' uniciclo attraverso una matlab function alla quale abbiamo dato come input la "road distance", sulla base di quest'ultima, campionando l'intero tracciato (soffermandoci soprattutto nelle curve più critiche come quelle a gomito) e procedendo per tentativi abbiamo determinato le velocità che permettono al veicolo di percorrere la curva nel modo migliore.



```
1. function v = speeddemand(rd)
2. %#codegen
3.
4. persistent v1 ;
5. if isempty(v1)
6.     v1=7;
7. end
8. if rd>40 && rd<80
9.     v1=45;
10. end
11. if rd>80 && rd<150
12.     v1=85;
13. end
14.
15. if rd>318 && rd<330
16.     v1=53;
17. end
18.
19. if rd>475 && rd<515
20.     v1=33;
21. end
22.
23. if rd>580 && rd<910
24.     v1=60;
25. end
26.
27. if rd>910 && rd<1080
28.     v1=37;
29. end
30.
31.
32. if rd>1080 && rd<1223 ecc.....
```

## Controllo del cambio

Per gestire il cambio del veicolo abbiamo utilizzato, per le prime 2 marce un controllo sulla velocità, mentre per le marce successive abbiamo creato un controllo sui giri del motore al minuto. Abbiamo utilizzato un controllo basato sugli stati del cambio in modo tale da avviare ai tick di aggiornamento del processore e per lo stesso motivo abbiamo dovuto usare la velocità come controllo delle prime due marce.



```
1. function gear = gear(rpm, v)
2.
3. statos1=0;
4. statod1=0;
5. rpml=0;
6. v=v(1);
7. persistent gear1 statos statod;
8. if isempty(gear1)
9.     gear1= int32(1);
10.    statos = 0;
11.    statod= 0;
12. end
13. if v<18 && v>0
14.    gear1=int32(1);
15. end
16. if v>18 && v<27
17.    gear1=int32(2);
18. end
19.
20. if rpm < 16000
21.    statos =1;
22. end
23. if rpm > 9500
24.    statod =1;
25. end
26.
27. if rpm > 16000 && statos == 1 && gear1<7 && gear1>1
28.    gear1= gear1+1;
29.    statos=0;
30. end
31. if rpm<9500 && statod == 1 && gear1>2
32.    gear1= gear1-1;
33.    statod=0;
34. end
35.
36. gear = gear1;
```

# I TISCHITOSCHI