

# Valutazione Real-Time del Contatto Pneumatico/Strada con Algoritmi Dedicati

Relatore:

**Prof. Enrico Bertolazzi**

Università di Trento

Candidato:

**Davide Stocco**

Co-relatore:

**Dott. Ing. Matteo Ragni**

AnteMotion S.r.l



## Motivazioni della Tesi

### ① Simulatore con

- *Software in the Loop* (SIL)
- *Hardware in the Loop* (HIL)
- *Driver in the Loop* (DIL)

per la validazione degli *Advanced Driver-Assistance Systems* (ADAS)

### ② Valutazione del contatto pneumatico/strada

# Obiettivi della Tesi

- 1 Sviluppo di una libreria C++ per la valutazione del contatto pneumatico/strada
- 2 Applicazione in tempo reale



# Intersezione pneumatico/*mesh*

- 1 Analisi sintattico-grammaticale del formato *rdf*
- 2 Istanziamento della *mesh*
- 3 Istanziamento dello pneumatico
- 4 Posizionamento dello pneumatico nello spazio
- 5 Intersezione degli alberi AABB per trovare i triangoli candidati
- 6 Scelta del modello di contatto
- 7 Utilizzazione di algoritmi di tipo geometrico per valutare il contatto
- 8 Estrazione dei risultati

# Il formato rdf per le superfici stradali

---

[NODES]

```
{ id x_coord y_coord z_coord }  
0 2.64637 35.8522 -1.59419e-005  
1 4.54089 33.7705 -1.60766e-005  
2 4.52126 35.8761 -1.62482e-005  
3 2.66601 33.7456 -1.57714e-005  
4 0.771484 35.8282 -1.56367e-005  
... ..
```

[ELEMENTS]

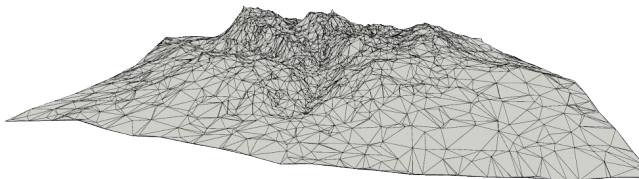
```
{ n1 n2 n3 mu }  
1 2 3 1.0  
2 1 4 1.0  
... ..
```

---

# Analisi sintattico-grammaticale del formato rdf

- 1 Estrazione dei [NODES]
- 2 Estrazione degli [ELEMENTS]
- 3 Istanziamento dei triangoli componenti la *mesh*

!!! Non esiste uno *standard* per questo formato

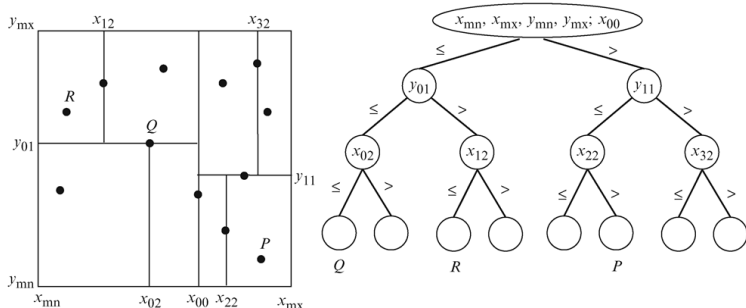


## Istanziamento della *mesh*

```
TireGround::RDF::MeshSurface Road(  
    "./file.rdf" // Path to the *.rdf file  
);
```

# Albero delle *Axis-Aligned Bounding Boxes* (AABB)

- 1 Raggruppamento ricorsivo delle AABB dei triangoli della *mesh*
- 2 Diminuzione in scala logaritmica del numero di comparazioni
- 3 Solo confronti logici



# Modellizzazione geometrica dello pneumatico

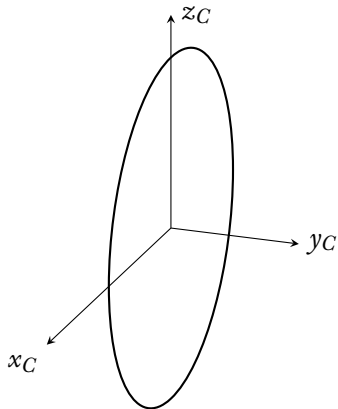
Ente normatore: *European Tire and Rim Technical Organization* (ETRTO)



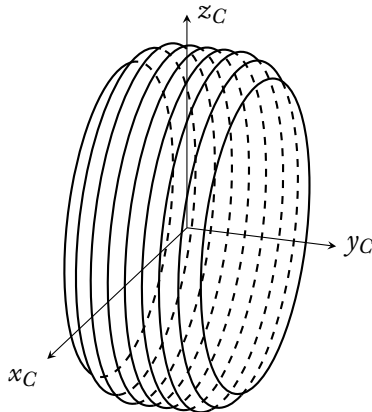


# Rappresentazione dello pneumatico tramite dischi

- 1 Uno o più dischi indeformabili
- 2 Movimenti relativi consentiti

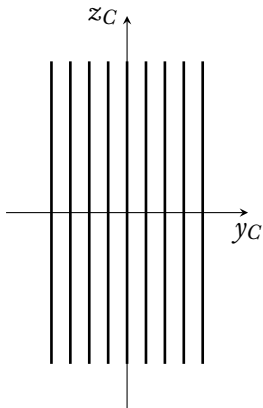


Disco singolo

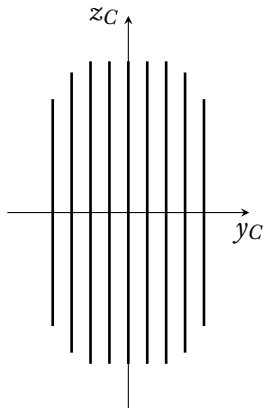


Dischi multipli

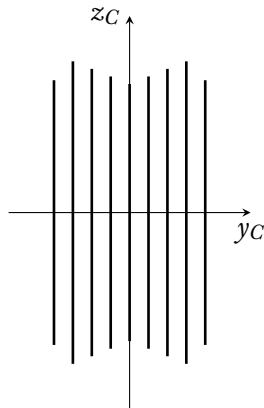
# Disposizione dei dischi



Raggio uniforme



Spalla raccordata



Profilo personalizzato

# Istanziamento dello pneumatico

---

```
TireGround::Tire* TireSD = new TireGround::MagicFormula(  
    SectionWidth, // [m]  
    AspectRatio,  // [%]  
    RimDiameter,  // [in]  
    SwitchNumber  // Max triangles in the shadow  
);
```

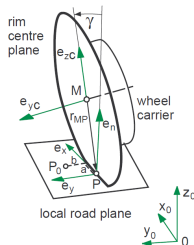
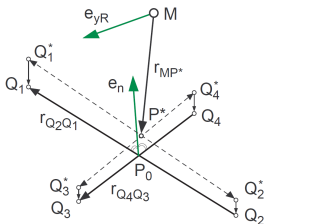
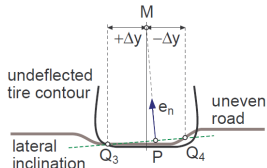
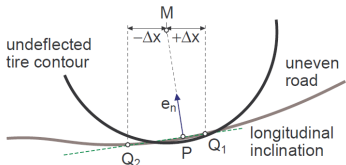
---

```
TireGround::Tire* TireMD = new TireGround::MultiDisk(  
    SectionWidth, // [m]  
    AspectRatio,  // [%]  
    RimDiameter,  // [in]  
    RadiusVec,    // Disks radius vector [m]  
    PointsNumber, // Sampling points for each disk  
    SwitchNumber  // Max triangles in the shadow  
);
```

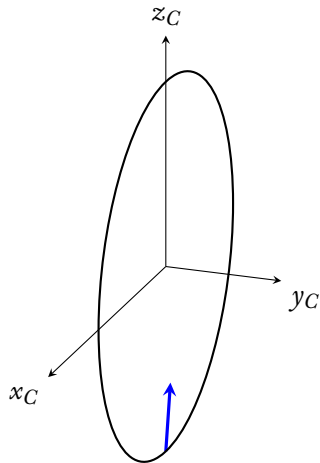
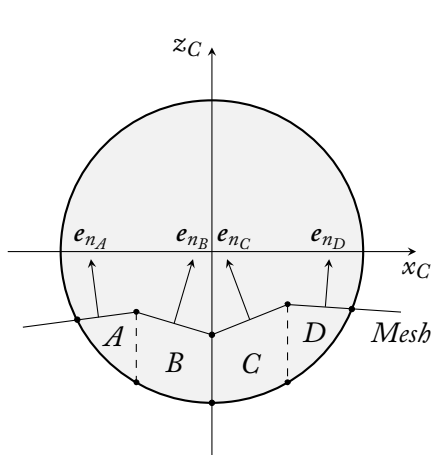
---

## Modelli di contatto per pneumatico mono-disco

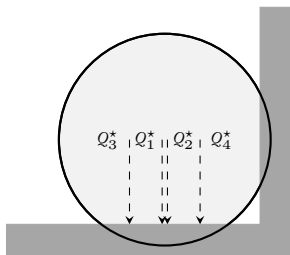
### 1 Modello di contatto di Rill



## 2 Modello di contatto ponderato in base all'area d'intersezione

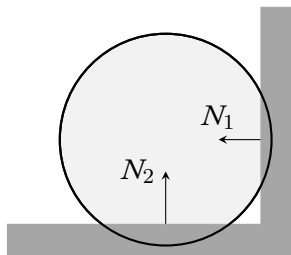


# Differenze tra le due tipologie di modelli di contatto



Rill:

- Non rileva ostacoli frontali
- Approssimativo

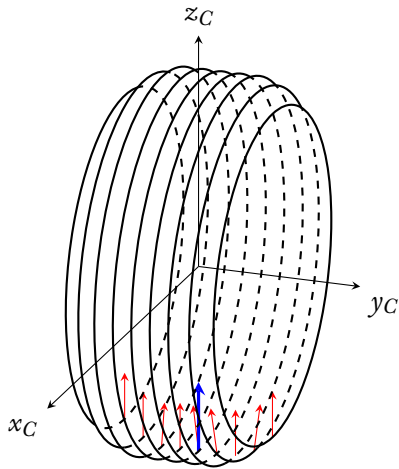
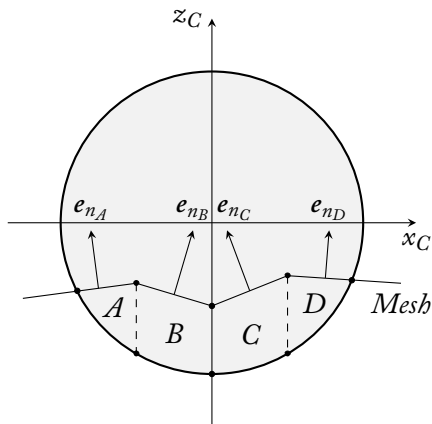


Ponderato sull'area d'intersezione:

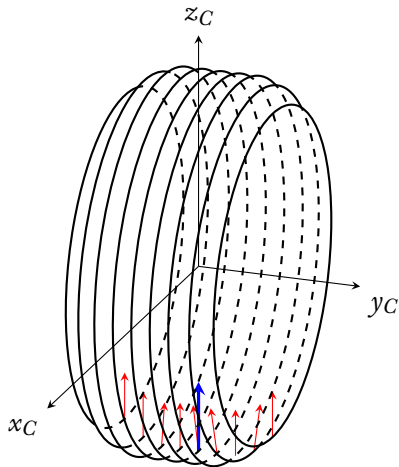
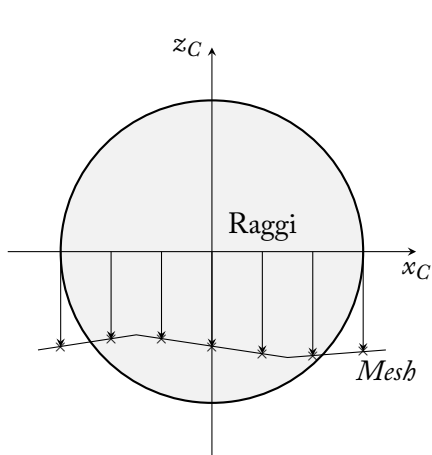
- + Rileva ostacoli frontali
- + Robusto se la *mesh* ha "buchi"
- Poco robusto se i triangoli sono sovrapposti

# Modelli di contatto per pneumatico multi-disco

## ① Modello di contatto ponderato in base all'area d'intersezione



## 2 Modello di contatto tramite campionamento





# Posizionamento dello pneumatico nella libreria

---

```
bool Out = SampleTire->setup(  
    Road, // Superficie stradale  
    TM    // Matrice di trasformazione 4x4  
);
```

---

```
bool Out = SampleTire->setup(  
    Normal,    // Vettore normale al piano  
    Point,     // Punto appartenente al piano  
    Friction,  // Coefficiente di attrito nel piano  
    TM        // Matrice di trasformazione 4x4  
);
```

---

# Estrazione dei risultati

---

```
TireGround::vec3 N;  
TireGround::vec3 P;  
TireGround::real_type Friction;  
TireGround::real_type Rho;  
TireGround::real_type RhoDot;  
TireGround::real_type RelativeCamber;  
TireGround::real_type Area;  
TireGround::real_type Volume;
```

```
SampleTire->getNormal(N);  
SampleTire->getMFpoint(P);  
SampleTire->getFriction(Friction);  
SampleTire->getRho(Rho);  
SampleTire->getRhoDot(PreviousRho,TimeStep,RhoDot);  
SampleTire->getRelativeCamber(RelativeCamber);  
SampleTire->getArea(Area);  
SampleTire->getVolume(Volume);
```

---

# Prestazioni della libreria

Pneumatico 250/55R11

Campionamenti = 30000

	Modello di contatto	
	Ponderato sull'area	<i>Mix</i>
$T_{step} [\mu s]$	9.6688	9.7658
$\sigma^2 [\mu s^2]$	1.4018	1.4983

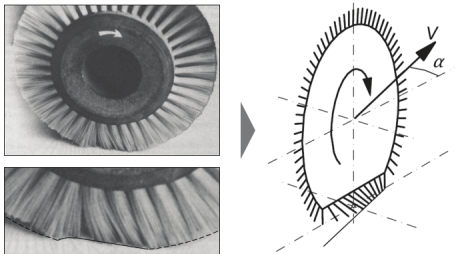
*Switch* Area ▷ Rill a 10 triangoli

	Precisione		Modello di contatto	
	Dischi	Punti	Ponderato sull'area	<i>Mix</i>
$T_{step} [\mu s]$	5	5	24.5736	39.6069
$\sigma^2 [\mu s^2]$	10	5	42.6262	439.6915
$T_{step} [\mu s]$	5	10	24.6686	55.7135
$\sigma^2 [\mu s^2]$	10	10	41.4114	479.8682

*Switch* Area ▷ Campionamento a 10 triangoli

# Possibili sviluppi

- 1 Definizione di uno *standard* per il formato *rdf*
- 2 Implementazione di un *parser* sufficientemente efficiente e stabile
- 3 Rappresentazione dello pneumatico mediante un modello fisico



# FINE

# Intersezione trà alberi di tipo AABB

# Intersezione trà alberi di tipo AABB

Volendo intersecare due semplici AABB, quali:

$$A = [A.minX, A.maxX, A.minY, A.maxY]$$

$$B = [B.minX, B.maxX, B.minY, B.maxY]$$

verrà usata la seguente funzione:

---

```
function intersect(A,B) {  
    return (A.minX <= B.maxX && A.maxX >= B.minX) &&  
    (A.minY <= B.maxY && A.maxY >= B.minY)  
}
```

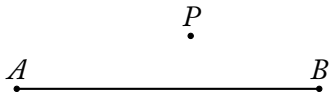
---

# Intersezione trà entità geometriche



# Intersezione segmento-punto

Dato un punto  $P = (x_p, y_p)$  e un segmento definito dai punti  $A = (x_A, y_A)$  e  $B = (x_B, y_B)$ .

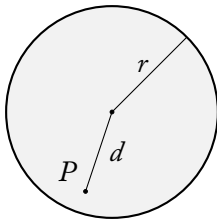


Per determinare se il punto  $P$  è interno al segmento gli *step* sono:

- 1 creazione di un vettore  $\vec{AB}$  e di un vettore  $\vec{AP}$ ;
- 2 calcolo del prodotto vettoriale  $\vec{AB} \times \vec{AP}$ , se il modulo del vettore risultante è nullo allora il punto  $P$  appartiene al segmento considerato;
- 3 calcolo del prodotto scalare tra  $\vec{AB}$  e  $\vec{AP}$ , se è nullo allora  $P \equiv A$ , se è pari al modulo di  $\vec{AB}$  allora  $P \equiv B$ , se è compreso tra 0 il modulo di  $\vec{AB}$ , allora il punto  $P$  giace all'interno del segmento considerato.

# Intersezione punto-cerchio

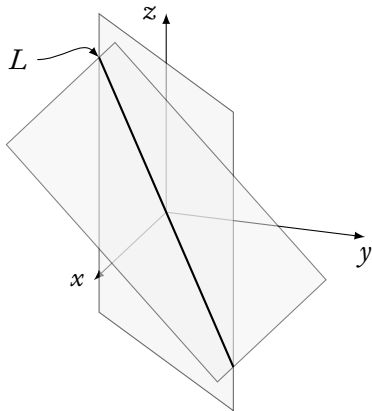
Data una circonferenza con centro  $C = (x_c, y_c)$  e raggio  $r$ , il problema consiste nel trovare se un generico punto  $P = (x_p, y_p)$  è all'interno, all'esterno o corrispondente alla circonferenza.



La soluzione al problema è semplice: la distanza tra il centro della circonferenza  $C$  e il punto  $P$  è data dal teorema di Pitagora, ovvero:

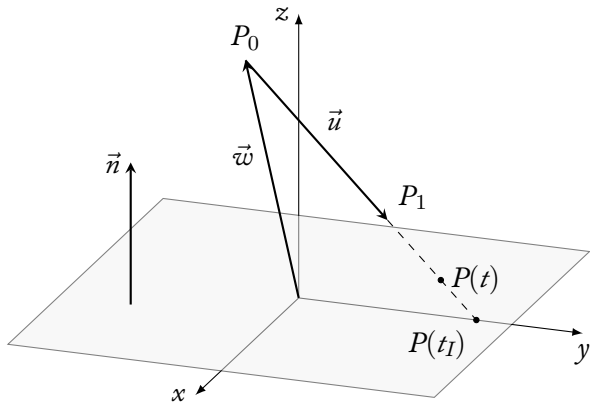
$$d = \sqrt{(x_p - x_c)^2 + (y_p - y_c)^2}$$

# Intersezione piano-piano



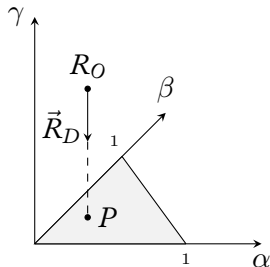
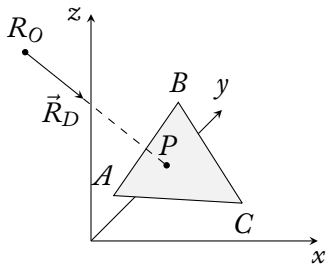
$$L(s) = \frac{(d_2 \vec{n}_1 - d_1 \vec{n}_2) \times \vec{u}}{|\vec{u}|^2} + s\vec{u}$$

# Intersezione piano-segmento e piano-raggio



$$\vec{n} \cdot (\vec{w} + t\vec{u}) = 0 \quad t_I = -\frac{\vec{n} \cdot \vec{w}}{\vec{n} \cdot \vec{u}}$$

# Intersezione raggio-triangolo

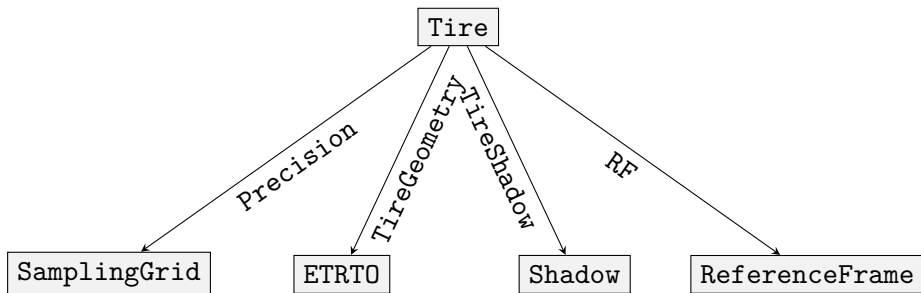


$$R_O + t\vec{R}_D = A + u(B - A) + v(C - A)$$

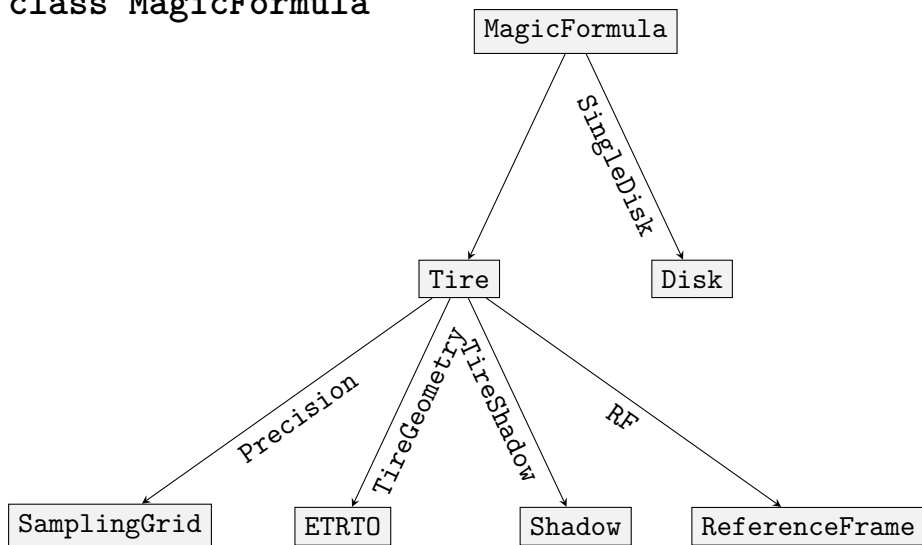
$$\begin{bmatrix} t \\ u \\ v \end{bmatrix} = \frac{1}{(D \times E_2) \cdot E_1} \begin{bmatrix} (T \times E_1) \cdot E_2 \\ (D \times E_2) \cdot T \\ (T \times E_1) \cdot D \end{bmatrix} = \frac{1}{P \cdot E_1} \begin{bmatrix} Q \cdot E_2 \\ P \cdot T \\ Q \cdot D \end{bmatrix}$$

# La libreria TireGruond

```
class Tire
```



```
class MagicFormula
```





```
class MultiDisk
```

