

# Specifiche FSM F1/10

(03/07/2023)

## Supervisor node (Luca Inghilterra)

Il nodo di supervisione ha il compito di immagazzinare lo stato di salute generale del veicolo tramite una macchina a stati finiti, che sarà implementata tramite l'uso di una libreria a scelta tra *SMACC2* ([ROS2 package](#), [github](#)) e *YASMIN* ([github](#)). L'output del nodo è lo stato della FSM e dovrà essere pubblicato come topic. Per simulare i guasti è possibile utilizzare un nodo di prova ed utilizzare le funzionalità di [deadline/liveliness](#) fornite dal framework ROS2.

### Definizioni:

- Stack di guida principale: Soluzione di guida autonoma performante ma soggetta a guasti o condizioni potenzialmente pericolose
- Stack di guida secondario: Soluzione di guida autonoma meno performante ma più affidabile e conservativa nella guida
- Guasto comune: Condizione tale per cui la guida autonoma con lo stack principale non può proseguire
  - Un nodo critico diventa non responsivo (manca un certo numero di deadline)
  - Viene identificata una situazione di rischio per il veicolo (perdita di aderenza, comandi di guida che portano a collisione certa, etc...)
- Guasto grave: Un componente hardware diventa non responsivo
  - Guasto a driver di sensori o attuatori

### Il nodo implementa i seguenti stati:

- **Idle [I]**: Il nodo è attivo e attende segnali dall'esterno
- **Manual [M]**: Il veicolo è in stato di guida manuale
  - In questo stato non vengono effettuati controlli sui guasti
  - Tutti i comandi di guida forniti da stack principale e secondario sono ignorati
- **Active [A]**: Il veicolo è in stato di guida autonoma
  - In questo stato vengono effettuati controlli sui guasti
  - Il controllo è affidato allo stack di guida principale
- **Emergency Takeover [ET]**: Il veicolo si trova in uno stato di rischio
  - Il controllo è affidato allo stack di guida secondario
- **Emergency Stop [ES]**: Il veicolo non è in grado di marciare autonomamente
  - I comandi di guida vengono ignorati e il veicolo viene fermato sul posto

### Il nodo implementa le seguenti transizioni di stato:

- (I)  $\longleftrightarrow$  (M): Servizio richiamabile dall'esterno
- (M)  $\longleftrightarrow$  (A): Servizio richiamabile dall'esterno
- (A)  $\rightarrow$  (ET): Si verifica un guasto comune
- (ET)  $\rightarrow$  (A): Il guasto comune viene risolto
- (ET)  $\rightarrow$  (M): Servizio richiamabile dall'esterno
- (A, M, ET)  $\rightarrow$  (ES): Si verifica un guasto grave
- (ES)  $\rightarrow$  (ET): Il guasto grave viene risolto e si è entrati in stato di ES da A o ET
- (ES)  $\rightarrow$  (M): Il guasto grave viene risolto e si è entrati in stato di ES da M

## Strategy node (Leonardo Pedretti, Giulia Casarini)

Il nodo strategico implementa due macchine a stati finiti. La prima sarà esposta e implementata utilizzando la funzionalità [Managed Nodes](#) di ROS2. Questa prima macchina a stati finiti ha il compito di gestire il ciclo di vita del nodo, permettendo di configurarlo, attivarlo ed eventualmente disattivarlo dall'esterno. La seconda macchina a stati finiti è quella strategica, che sarà implementata tramite l'uso di una libreria per FSM a scelta tra SMACC2 ([package](#), [github](#)) e YASMIN ([github](#)).

Il nodo riceve in input dalla rete ROS2 la posizione del veicolo (di tipo Odometry) e un vettore con le posizioni degli eventuali avversari (di tipo PoseArray). Le callback ROS2 per questi topic dovranno essere gestite da più thread, utilizzando [MultiThreadedExecutor](#) e [CallbackGroup](#) di ROS2. Di conseguenza, sarà necessario proteggere le variabili condivise tra più thread con primitive di sincronizzazione. Si preveda inoltre l'inizializzazione del nodo tramite un parametro configurabile Dth. L'output del nodo è lo stato della FSM di strategia e dovrà essere pubblicato come topic.

L'ostacolo è definito dalle seguenti variabili:

- pose: posizione dell'ostacolo ricevuta dal messaggio ROS2
- $D_i$ : Distanza tra l'ostacolo e l'interno della pista
  - Per comodità può essere assegnato ad un valore casuale al momento dell'inizializzazione dell'ostacolo
- $D_o$ : Distanza tra l'ostacolo e il bordo esterno della pista
  - Per comodità può essere assegnato ad un valore casuale al momento dell'inizializzazione dell'ostacolo

Requisiti di implementazione per la Managed Nodes FSM

- *onConfigure()*: configurazione dei parametri operativi del nodo ( $D_{th}$ )
- *onCleanup()*: reset di tutti gli stati/configurazioni/variabili per una nuova esecuzione
- *onActivate()*: la FSM strategica viene lanciata insieme all'esecutore per le callback
- *onDeactivate()*: il nodo viene disattivato e le callback rilasciate

Per quanto riguarda la FSM strategica, il nodo implementa i seguenti stati:

- **Ready [R]**: il veicolo è fermo in attesa di istruzioni
- **Global [G]**: il veicolo segue la traiettoria globale
- **Follow [F]**: il veicolo segue l'avversario in attesa di un'opportunità di sorpasso
- **Overtake Inside [OI]**: il veicolo sta effettuando un sorpasso a destra
- **Overtake Outside [OO]**: il veicolo sta effettuando un sorpasso a sinistra

Transizioni

- (R) → (G): Servizio richiamabile dall'esterno
- (G) → (F): Esiste almeno un ostacolo
- (F) → (OI): L'ostacolo più vicino è al più a  $D_{th}$  metri di distanza e ha  $D_i \leq D_o$
- (F) → (OO): L'ostacolo più vicino è al più a  $D_{th}$  metri di distanza e ha  $D_o < D_i$
- (OI, OO) → (AO): L'ostacolo più vicino è a più di  $D_{th}$  metri di distanza
- (F, OI, OO) → (G): Non sono più presenti ostacoli
- (G, F, OI, OO) → (R): Servizio richiamabile dall'esterno
- Si preveda inoltre un limite minimo di  $T_{nc}$  secondi che dovranno trascorrere dal momento in cui si entra nello stato (F) al successivo cambio di stato verso (G).