

Dissuasore di velocità autostradale (DVA)

Il sistema proposto, denominato DVA, ha il compito di indurre i conducenti dei veicoli che percorrono un’autostrada a procedere ad una velocità corretta per il suddetto percorso, vale a dire a rispettare un limite minimo pari a 80 km/h ed un limite massimo di 130 km/h, oltre ad impedire l’accesso in tale strada ai veicoli non provvisti dei seguenti requisiti minimi:

- **Velocità di almeno 80 km/h;**
- **Peso di almeno 1,3 t;**
- **Cilindrata di almeno 250 cc.**

Tali informazioni sono rilevate attraverso l’utilizzo di tre sensori, chiamati, rispettivamente, “speed sensor”, “weight sensor” e “displacement sensor”, i quali, attraverso un “gateway”, spediranno i dati dalla rete wireless formata da questi quattro nodi ad una rete cablata, costituita da altri quattro nodi, denominati “infringement controller”, “fine controller”, “signage controller” e “gateway”, il cui compito è, rispettivamente, la generazione del livello di infrazione da 0 a 5, della multa, del segnale e dei punti associati al livello di infrazione e della scelta della segnaletica da mostrare al conducente del veicolo, ossia di colore verde fisso se non è stata commessa un’infrazione o giallo lampeggiante in caso contrario, e la ricezione dei pacchetti dalla rete precedente.

I. INTRODUZIONE

Percorrendo una tratta autostradale, può capitare che si riscontrino delle situazioni inaspettate e pericolose, dovute ad atteggiamenti errati da parte di alcuni conducenti di veicoli, quali, per esempio, mantenimento di una velocità superiore al limite massimo o inferiore al limite minimo, occupazione impropria di corsie dedicate al sorpasso o alla sosta e scarsa prudenza in presenza di tratti con restringimenti di carreggiata o lavori in corso. Tali problematiche si possono rilevare anche in altre tratte stradali, ma è all'interno di un'autostrada che hanno un'incidenza maggiore nel provocare rischi evitabili ed incidenti di gravità più o meno elevata.

Per limitare il verificarsi delle suddette situazioni, sono stati inseriti degli **autovelox** in appositi punti di queste tratte autostradali. Questi catturano le immagini delle targhe dei veicoli e spediscono al computer centrale quelle dei soli veicoli in contravvenzione. In genere, gli autovelox sono installati su un treppiedi mobile, su un'autovettura ferma a bordo strada o in una postazione fissa, chiamata autobox.

Una soluzione alternativa a quella vista finora può essere l'instaurazione di **dissuasori di velocità**. Questi sono un mezzo preventivo di sicurezza, utilizzato in punti stradali pericolosi, dove l'eccessiva velocità potrebbe causare incidenti o situazioni inaspettate ed insidiose. Sono costituiti da un lettore di velocità istantaneo che comunica con un display nel quale è mostrata tale velocità e, spesso, anche dei consigli utili, come ad esempio le sanzioni in cui si incorrerebbe se invece di un dissuasore fosse installato un autovelox.

Il sistema proposto, invece, è un **dissuasore di velocità autostradale**, più brevemente **DVA**, ed è costituito da un sensore di velocità, uno di peso ed uno di cilindrata, tutti alimentati da batterie. I dati rilevati dai precedenti sensori permettono di stabilire se il conducente di un veicolo è in contravvenzione, ed, eventualmente, di sancire il livello di infrazione, mostrando una segnaletica gialla lampeggiante ed esponendo su due display la possibile multa da pagare ed eventualmente il numero di punti sulla patente da sottrarre. In caso contrario, invece, la segnaletica sarebbe verde fissa e i display della multa e del numero di punti presenterebbero il valore di default 0.

La **sezione II** mostra i lavori simili al sistema presentato.

La **sezione III** mostra l'approccio proposto per la risoluzione del problema.

La **sezione IV** mostra lo scenario di funzionamento in ambiente simulato.

La **sezione V** mostra la valutazione delle performance del sistema DVA.

La **sezione VI** mostra le conclusioni tirate e gli sviluppi futuri.

II. RELATED WORKS

Come già detto in precedenza, i lavori simili al sistema proposto sono costituiti dagli autovelox e i dissuasori di velocità.

Tuttavia, per quanto riguarda gli **autovelox**, non sempre questi vengono mantenuti in condizioni tali da permetterne il corretto funzionamento. Infatti, capita spesso che questi siano presenti ma materialmente non svolgono il proprio dovere, e gli automobilisti, venuti a conoscenza di ciò, ad esempio, continuano a proseguire per quel percorso ad una velocità non rientrante nei limiti previsti, senza alcun timore riguardo possibili multe che potrebbero ricevere. Quindi, in una situazione del genere, è come se gli autovelox non fossero installati, perciò non vi è nessun controllo sulla percorrenza corretta dei veicoli in autostrada. Le precedenti considerazioni sono valide anche quando, all'interno di un autobox, non vi è alcun autovelox, circostanza che si verifica abbastanza frequentemente.

Per ciò che concerne i **dissuasori di velocità**, visto che il loro scopo non è sanzionatorio, ma preventivo ed educativo, questi non comportano sanzioni. E' importante, però, il messaggio che questi strumenti vogliono lanciare, vale a dire quello per cui si avvisa l'automobilista del fatto che procedeva ad una velocità troppo eccessiva e la prossima volta, invece di un semplice dissuasore, potrebbe esserci un autovelox pronto a filmare il suo veicolo o potrebbe trovarsi in situazioni molto insidiose. Ma può capitare che queste installazioni siano ignorate dagli automobilisti, in quanto considerate come non pericolose, data la mancanza di emissione di sanzioni, sminuendo così i rischi stradali a cui si può andare incontro. Inoltre, questi strumenti sono concentrati maggiormente in strade extraurbane e all'ingresso di centri abitati, dunque nei tratti in cui le velocità dei veicoli sono molto alte, quali quelli autostradali, non vi sono elevati livelli di controllo e sicurezza.

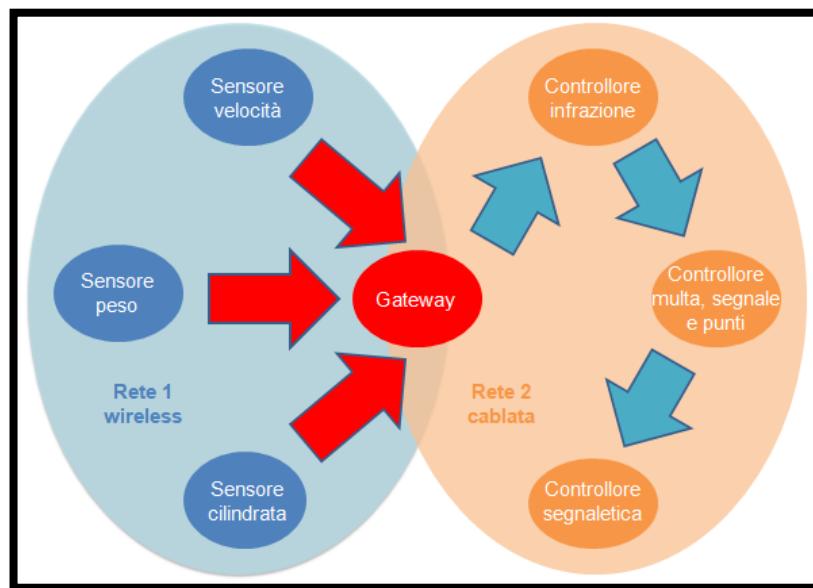
Invece, il sistema **DVA** è pur sempre un dissuasore di velocità, ma impiegato in quelle zone di percorrenza dei veicoli ad alto tasso di presenza di situazioni rischiose, che risulta essere fondamentale per impedire l'accesso ai mezzi non provvisti dei requisiti minimi elencati in precedenza, grazie all'azione dei sensori di peso e di cilindrata di cui è costituito, oltre che per rilevare se un veicolo procede non rispettando i limiti autostradali, sia quello minimo sia quello massimo, attraverso l'attività svolta dal classico sensore di velocità.

III. APPROCCIO PROPOSTO

Per il funzionamento del sistema **DVA**, si utilizzano due reti:

- Una wireless, per la rilevazione dei dati, costituita dai sensori di velocità, peso e cilindrata e dal gateway;
- Una cablata, per la generazione dei messaggi da mostrare all'automobilista, composta da controllori e gateway.

I sensori utilizzano la comunicazione wireless a basso consumo energetico 802.15.4 (ZigBee), mentre i controllori usufruiscono della comunicazione cablata CSMA/CD (Ethernet). La **topologia del sistema** è la seguente:



I task sensori prevedono due file:

- Di inizializzazione, simile per tutti i tre sensori e costituito dalla successiva sequenza di operazioni:
 - Inizializzazione del kernel tramite la funzione TrueTime “ttInitKernel” a “prioFP”, per usufruire di uno scheduling a priorità fissata;
 - Configurazione del consumo energetico del kernel alimentato a batteria attraverso la funzione TrueTime “ttSetKernelParameter” a 10 mW;
 - Pulizia dello spazio di indirizzamento in memoria della variabile persistente mediante il comando “Clear” della funzione associata;
 - Configurazione del task periodico utilizzato per la rilevazione effettuata dal componente fisico per mezzo della funzione TrueTime “ttCreatePeriodicTask”, a cui vengono passati come argomenti, nel seguente ordine, il nome del task, il tempo di start, il periodo di funzionamento del task ed il nome della funzione associata;
 - Impostazione della priorità del task grazie alla funzione TrueTime “ttSetPriority”, alla quale vengono passati come argomenti la priorità espressa sotto forma di valore numerico ed il nome del task.
- Di funzione, anche in questo caso simile per tutti i tre sensori e formato dalla creazione di una variabile persistente e dalle seguenti tre porzioni operative:
 - La prima è quella in cui si effettua:
 - ★ L'inizializzazione al primo avvio della variabile persistente, posta uguale a 0;
 - ★ La memorizzazione nella struttura “info” del dato ricevuto dal canale analogico 1, tramite la funzione TrueTime “ttAnalogIn”;
 - La seconda è quella nella quale si ha:
 - ★ La memorizzazione nella struttura “type” di un tipo di segnale;
 - ★ La memorizzazione nella struttura “time” dell'istante corrente;
 - ★ La scrittura della struttura “info” sul canale analogico 1;
 - ★ L'invio del dato contenente tutte queste strutture al gateway della rete 1.
 - La terza è quella in cui si svolge:

- ★ L'incremento il valore della variabile persistente;
- ★ L'assegnazione di tale valore alla struttura "packet";
- ★ La memorizzazione nella struttura "type" un tipo di segnale diverso da quello della precedente porzione;
- ★ La scrittura della struttura "packet" sul canale analogico 2;
- ★ L'invio del dato avente tutte queste strutture al gateway della rete 1.

I **task controllori** prevedono, invece, tre file:

- Di inizializzazione, simile per tutti i tre controllori e costituito dalla successiva sequenza di operazioni:
 - Inizializzazione del kernel tramite la funzione TrueTime "ttInitKernel" a "prioFP", per usufruire di uno scheduling a priorità fissata;
 - Creazione di differenti mailbox, riguardanti: velocità, peso, cilindrata, infrazione, multa, segnaletica d'avviso e punti della patente;
 - Configurazione del task periodico utilizzato per la rilevazione effettuata dal componente fisico per mezzo della funzione TrueTime "ttCreatePeriodicTask", a cui vengono passati come argomenti, nel seguente ordine, il nome del task, il tempo di start, il periodo di funzionamento del task ed il nome della funzione associata;
 - Impostazione della priorità del task grazie alla funzione TrueTime "ttSetPriority", alla quale vengono passati come argomenti la priorità espressa sotto forma di valore numerico ed il nome del task;
 - Realizzazione dell'handler, di cui vengono definiti nome, priorità e relativa funzione, ad una rete, attraverso la funzioni TrueTime "ttCreateHandler", a cui vengono passati come argomenti i tre elementi definiti sopra;
 - Collegamento dell'handler ad una rete mediante la funzione TrueTime "ttAttachNetworkHandler", alla quale si passano come argomenti il numero della rete ed il nome dell'handler.
- Di funzione, pure in questo caso simile per tutti i tre controllori e formato da due porzioni, di cui una operativa in cui si ha:
 - La ricezione di dati dalle rispettive mailbox, tramite la funzione TrueTime "ttTryFetch", o dai canali analogici, mediante la funzione TrueTime "ttAnalogIn";
 - La verifica che questi dati non siano vuoti;
 - L'invio di dati parziali prodotti e messaggi agli altri nodi della rete 2, per generare alla fine le informazioni da mostrare agli automobilisti. Più precisamente, i dati passano dal controllore dell'infrazione a quello di multa, segnale e punti, e da questo a quello della segnaletica.
- Di gestione della rete, in cui si esegue:
 - La ricezione di un messaggio dalla rete 2, tramite la funzione TrueTime "ttGetMsg";
 - L'invio di un messaggio ad una delle mailbox create precedentemente, mediante la funzione TrueTime "ttTryPost".

Il **gateway** prevede, anch'esso, tre file come quelli dei task appena descritti:

- Di **inizializzazione**, costituito dalla successiva sequenza di operazioni:
 - Inizializzazione del kernel tramite la funzione TrueTime "ttInitKernel" a "prioFP", per usufruire di uno scheduling a priorità fissata;
 - Configurazione del consumo energetico del kernel alimentato a batteria attraverso la funzione TrueTime "ttSetKernelParameter" a 10 mW;
 - Creazione di sei mailbox, due a testa per velocità, peso e cilindrata, di cui una per quando i pacchetti vengono mandati correttamente ed una per quando questi, invece, vengono persi;
 - Pulizia dello spazio di indirizzamento in memoria della variabile persistente mediante il comando "Clear" della funzione associata;
 - Configurazione del task non periodico utilizzato per la rilevazione effettuata dal componente fisico per mezzo della funzione TrueTime "ttCreateTask", a cui vengono passati come argomenti, nel seguente ordine, il nome del task, la scadenza del task ed il nome della funzione associata;

- Impostazione della priorità del task grazie alla funzione TrueTime “ttSetPriority”, alla quale vengono passati come argomenti la priorità espressa sotto forma di valore numerico ed il nome del task;
- Realizzazione dell’handler, di cui vengono definiti nome, priorità e relativa funzione, ad una rete, attraverso la funzioni TrueTime “ttCreateHandler”, a cui vengono passati come argomenti i tre elementi definiti sopra;
- Collegamento dell’handler ad una rete mediante la funzione TrueTime “ttAttachNetworkHandler”, alla quale si passano come argomenti il numero della rete ed il nome dell’handler.
- Di **funzione**, formato da tre porzioni, di cui due operative. Di queste ultime:
 - La prima è quella in cui si effettua:
 - ★ La realizzazione di sei variabili persistenti, due a testa per velocità, peso e cilindrata, di cui una conta i pacchetti ricevuti e l’altra i pacchetti persi;
 - ★ La verifica che quelli di quest’ultima tipologia non siano vuoti;
 - ★ La ricezione dei dati dalle mailbox per pacchetti su velocità, peso e cilindrata persi, tramite la funzione TrueTime “ttTryFetch”;
 - ★ La verifica che anche questi ultimi dati non siano vuoti;
 - ★ La scrittura su tre canali diversi della struttura “packet”, della variabile persistente che conta i pacchetti persi, dopo aver verificato se è necessario incrementarla, e della differenza tra l’istante corrente e la struttura “time”. Ciò deve essere effettuato per ognuno degli ultimi tre dati ricevuti dalle mailbox per pacchetti persi.
 - La seconda è quella nella quale si ha:
 - ★ La ricezione dei dati dalle mailbox per pacchetti su velocità, peso e cilindrata ricevuti, tramite la funzione TrueTime “ttTryFetch”;
 - ★ La verifica che questi ultimi dati non siano vuoti;
 - ★ L’invio di tali dati al controllore dell’infrazione della rete 2, qualora questi non dovessero risultare vuoti.
- Di **gestione della rete**, in cui si esegue:
 - La ricezione di un messaggio dalla rete 1, tramite la funzione TrueTime “ttGetMsg”;
 - L’invio di un messaggio ad una delle mailbox create precedentemente, mediante la funzione TrueTime “ttTryPost”;
 - L’avvio del task del gateway, attraverso la funzione TrueTime “ttCreateJob”.

Il **controllore a logica fuzzy** è costituito da tre input, un output e nove regole e sfrutta il modello “Sugeno”. Nello specifico:

- I tre **input** sono chiamati “Speed”, “Weight” e “Displacement”:
 - “Speed” ha un range da 0 a 220. Possiede sei livelli, denominati “Low”, “Correct”, “Little_High”, “Somewhat_High”, “Quite_High” e “Very_High”, rispettivamente con range da -10 a 80, da 80 a 135, da 135 a 147, da 147 a 178, da 178 a 200 e da 200 a 230, per rappresentare un valore di velocità basso, corretto, leggermente alto, alquanto alto, piuttosto alto e molto alto;
 - “Weight” ha un range da 0 a 20. Possiede tre livelli, denominati “Low”, “Medium” e “High”, rispettivamente con range da -1 a 1.3, da 1.3 a 3.5 e da 3.5 a 21, per rappresentare un valore di peso basso, medio e alto;
 - “Displacement” ha un range da 0 a 3000. Possiede due livelli, denominati “Low” e “Correct”, rispettivamente con range da -100 a 250 e da 250 a 3100, per rappresentare un valore di cilindrata basso e corretto.
- L’**output** è chiamato “Infringement”. Ha un range da 0 a 5. Possiede sei livelli, denominati “No_Fine”, “No_Circulate”, “Low”, “Medium”, “High” e “Very_High”, a cui rispettivamente sono associati i valori 0, 1, 2, 3, 4 e 5, per rappresentare un livello di infrazione nullo e tale da non generare una multa, molto basso e tale da impedire la circolazione del mezzo in autostrada, basso, medio, alto e molto alto;
- Le nove **regole** sono:
 - “Speed” = “Low” OR “Weight” = “Low” OR “Displacement” = “Low” producono “Infringement” = “No_Circulate”;

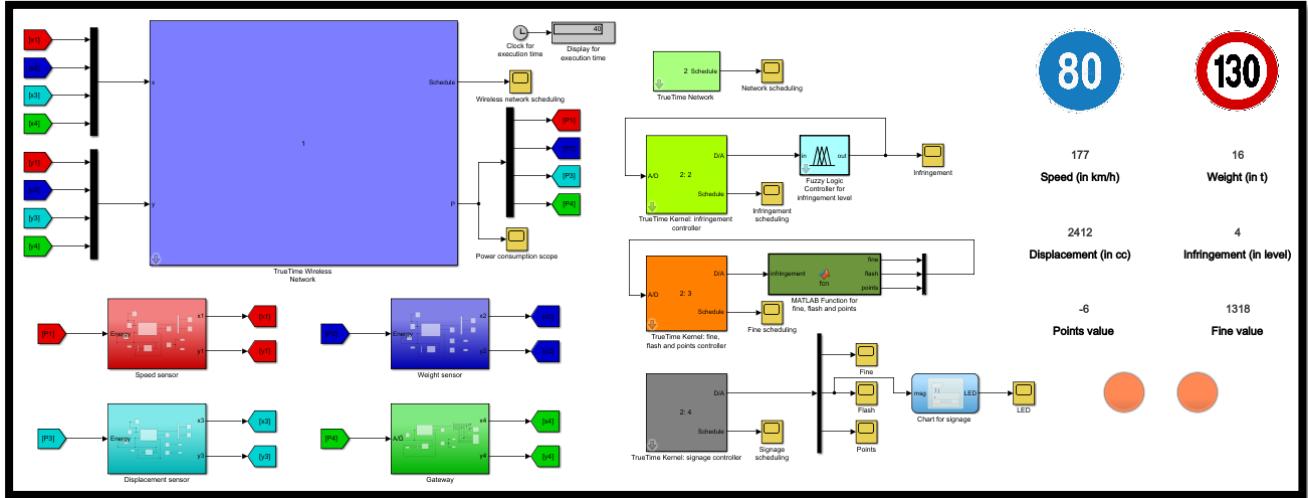
- “Speed” = “Correct” AND “Weight” != “Low” AND “Displacement” = “Correct” producono “Infringement” = “No_Fine”;
- “Speed” = “Correct” AND “Weight” = “Medium” producono “Infringement” = “No_Fine”;
- “Speed” = “Correct” AND “Weight” = “High” producono “Infringement” = “No_Fine”;
- “Speed” = “Correct” AND “Displacement” = “Correct” producono “Infringement” = “No_Fine”;
- “Speed” = “Little_High” produce “Infringement” = “Low”;
- “Speed” = “Somewhat_High” produce “Infringement” = “Medium”;
- “Speed” = “Quite_High” produce “Infringement” = “High”;
- “Speed” = “Very_High” produce “Infringement” = “Very_High”.

La **funzione MATLAB** possiede un input e tre output:

- L'**input** è chiamato “infringement” ed è costituito dal livello di infrazione prodotto dal controllore denominato “infringement controller”, mediante la logica fuzzy, ed ottenuto dal controllore definito “fine controller”. Possiede un valore da 0 a 5;
- I tre **output** sono chiamati “fine”, “flash” e “points” e rappresentano il quantitativo della multa da pagare, il valore associato alla segnaletica da mostrare all’automobilista ed il numero di punti della patente da rimuovere a quest’ultimo. Nella funzione sono subito inizializzati a 0, e, in base al valore assunto dall’input “Infringement”, possono assumere i seguenti valori:
 - Rimangono tutti pari a 0 quando “infringement” = 0;
 - “fine” = 103, “flash” = 1 e “points” = -2 quando “infringement” = 1;
 - “fine” = 105, “flash” = 1 e “points” = 0 quando “infringement” = 2;
 - “fine” = 421, “flash” = 1 e “points” = -3 quando “infringement” = 3;
 - “fine” = 1318, “flash” = 1 e “points” = -6 quando “infringement” = 4;
 - “fine” = 2054, “flash” = 1 e “points” = -10 quando “infringement” = 5.

IV. SCENARIO

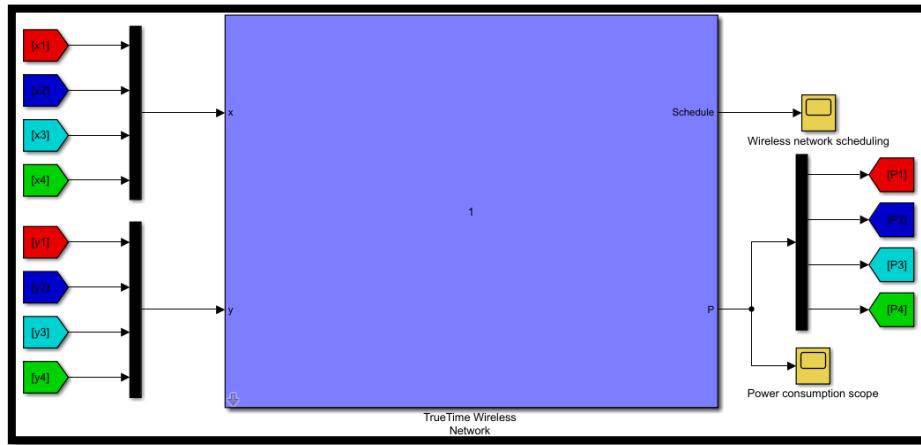
Il sistema DVA in ambiente simulato è stato sviluppato attraverso un approccio ingegneristico personale, non discriminante da una possibile realizzazione pratica. Il grafico complessivo del sistema è il seguente:



Come già esposto in precedenza, si hanno una rete con i sensori, a sinistra, ed una rete con i controllori, al centro. La simulazione in ambiente MATLAB del sistema DVA ha una durata di 40 secondi, tempo necessario per mostrare i risultati di tutti i possibili livelli di “infringement” descritti in precedenza. Inoltre, si sono resi necessari due diversi livelli di sviluppo.

Il primo livello di sviluppo, che riguarda la rete wireless 1, i sensori ed il gateway, è stato quello in cui si sono definiti i relativi comportamenti, rendendo i dati prossimi ad una realtà operativa congrua.

La struttura che realizza la rete 1 è la seguente:

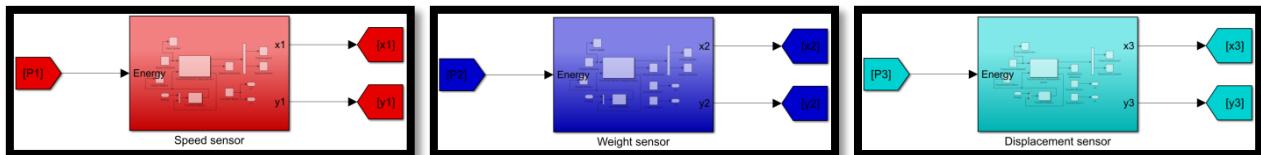


Come si nota da quest'ultima immagine, tale struttura è formata da:

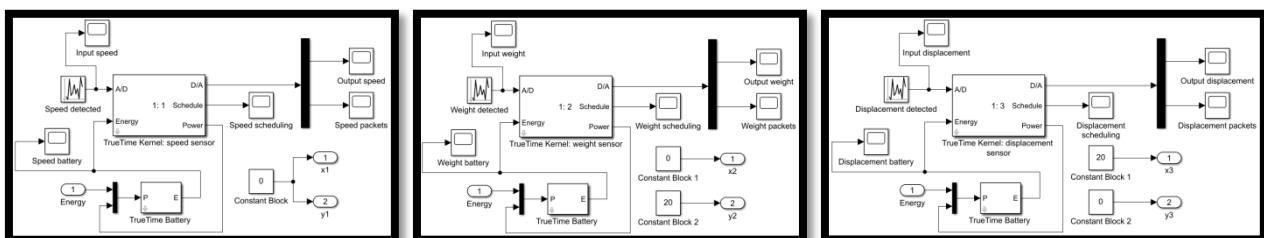
- Una Wireless Network (TrueTime Wireless Network), rappresentante la creatrice della rete 1. Essa ha i primi tre parametri impostati nella seguente maniera: “Network type” coincidente con il protocollo 802.15.4 (ZigBee), “Network Number” pari a 1, che stabilisce che la rete realizzata è la numero 1, e “Number of nodes” posto uguale a 4, per sancire che i nodi di tale rete sono 4. Inoltre riceve in ingresso i segnali “x” e “y” delle posizioni dei nodi di tale rete, i quali sono, rispettivamente, il risultato del raggruppamento, attraverso il blocco “Mux”, dei blocchi “x1”, “x2”, “x3” e “x4” e dei blocchi “y1”, “y2”, “y3” e “y4”. Non ha uscite, ma mostra le porte di scheduling (Schedule output port) e di consumo energetico (Power consumption output port);
- Otto blocchi di invio di segnali (Goto), con tag “x1”, “x2”, “x3” e “x4” raggruppati in alto a sinistra e tag “y1”, “y2”, “y3” e “y4” raggruppati in basso a sinistra;

- Quattro blocchi di ricezione di segnali (From), con tag “P1”, “P2”, “P3” e “P4” raggruppati a destra e rappresentanti il consumo energetico della rete wireless per ogni suo nodo;
- Due blocchi di modulazione (Mux), che permettono di usare un'unica connessione per trasferire quattro informazioni a testa. Il primo raggruppa i segnali provenienti dai blocchi “x1”, “x2”, “x3” e “x4”, mentre il secondo raggruppa quelli derivanti dai blocchi “y1”, “y2”, “y3” e “y4”;
- Un blocco di demodulazione (Demux), che divide i dati trasferiti con un'unica connessione in quattro parti, destinate ai blocchi “P1”, “P2”, “P3” e “P4”;
- Due osservatori di risultati (Scope), che permettono di visualizzare lo scheduling ed il consumo di energia del blocco “TrueTime Wireless Network”.

Le strutture dei **sensori** sono praticamente identiche e sono le seguenti:



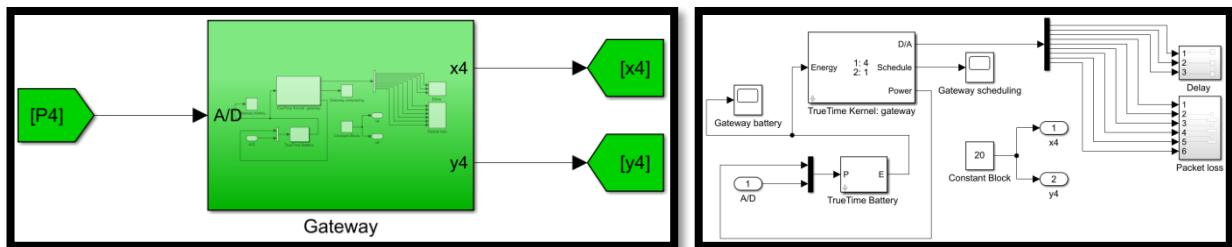
Più dettagliatamente, essi sono costruiti nella seguente maniera:



Come si evince considerando le sei immagini precedenti, ogni sensore è costituito da:

- Un kernel (TrueTime Kernel), rappresentante il vero e proprio nodo della rete 1. In particolare, il kernel del sensore della velocità è il nodo 1, quello del sensore del peso è il nodo 2 e quello del sensore della cilindrata è il nodo 3. Ogni kernel possiede un ingresso e due uscite. Per ogni kernel sono mostrate anche le porte di rifornimento energetico (Energy supply input port), di scheduling (Schedule output port) e di consumo energetico (Power consumption output port). La funzione passata al kernel del sensore di velocità è la “speedSensor_init”, quella trasferita al kernel del sensore di peso è la “weightSensor_init” e quella fornita al kernel del sensore di cilindrata è la “displacementSensor_init”;
- Un blocco esterno di ricezione di segnali (Goto), con tag “x1” e “y1” per il sensore di velocità, “x2” e “y2” per il sensore di peso e “x3” e “y3” per il sensore di cilindrata;
- Due generatori casuali di numeri (Uniform Random Number), con valore di “Maximum” pari a 220 per la velocità, 20 per il peso e 3000 per la cilindrata, e “Minimum” di valore 0, “Seed” uguale a 5 e “Sample time” equivalente a 2 per tutti e tre i sensori;
- Una porta di input (Inport), corrispondente alla porta 1 per tutti i sensori;
- Una batteria (TrueTime Battery), con “Initial energy” pari a 5 per ogni sensore;
- Dei blocchi di valore costante (Constant). In particolare, uno equivalente a 0 per la velocità, due, di cui uno pari a 0 ed uno uguale a 20, per il peso e due, dei quali uno di valore 20 ed uno posto a 0, per la cilindrata;
- Due porte di output (Outport), corrispondenti alla porta 1 ed alla porta 2 in tutti e tre i sensori;
- Un blocco di modulazione (Mux), che permette di usare un'unica connessione per trasferire due informazioni;
- Un blocco di demodulazione (Demux), che divide i dati trasferiti con un'unica connessione in due parti;
- Cinque osservatori di risultati (Scope), che permettono di visualizzare i valori di input, di output, lo scheduling, l'andamento dell'invio dei pacchetti e del livello di batteria in tutti i sensori.

Il **gateway** ha, invece, la seguente struttura, a sinistra quella generale e a destra quella dettagliata:

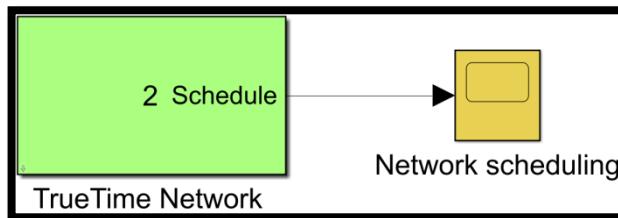


Come si vede dalle ultime due immagini, il gateway è formato da:

- Un kernel (TrueTime Kernel), rappresentante il nodo 4 della rete 1 ed il nodo 1 della rete 2. Il kernel non ha ingressi ma possiede nove uscite. Di questo sono mostrate anche le porte di rifornimento energetico (Energy supply input port), di scheduling (Schedule output port) e di consumo energetico (Power consumption output port). La funzione passatagli è la “gateway_init”;
- Un blocco esterno di ricezione di segnali (From), con tag “P4”;
- Due blocchi esterni di invio di segnali (Goto), con tag “x4” e “y4”;
- Dieci porte di input (Import), corrispondenti alla porta 1 per quella collegata alla batteria, alle porte 1, 2 e 3 nel sottosistema “Delay” ed alle porte 1, 2, 3, 4, 5 e 6 per il sottosistema “Packet loss”;
- Una batteria (TrueTime Battery), con “Initial energy” pari a 5;
- Due blocchi di valore costante (Constant), entrambi equivalenti a 20;
- Due porte di output (Outport), corrispondenti alla porta 1 ed alla porta 2;
- Un blocco di modulazione (Mux), che permette di usare un’unica connessione per trasferire due informazioni;
- Un blocco di demodulazione (Demux), che divide i dati trasferiti con un’unica connessione in nove parti;
- Otto osservatori di risultati (Scope), che permettono di visualizzare lo scheduling, i delay dei dati di velocità, peso e cilindrata e l’andamento dell’invio dei pacchetti per ogni sensore e del livello di batteria;
- Tre schermi (Display) per la visualizzazione del numero di pacchetti persi per la velocità, il peso e la cilindrata.

Il **secondo livello di sviluppo**, che riguarda la rete cablata 2 ed i controllori, è quello nel quale si sono elaborate le informazioni seguendo opportuni criteri.

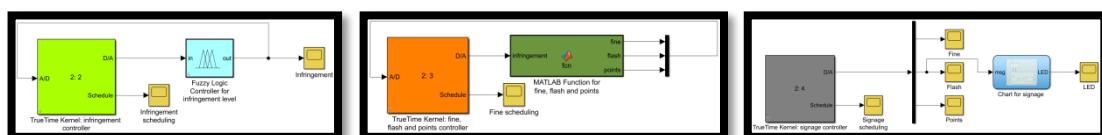
La struttura che realizza la rete 2 è la seguente:



Come si nota da quest’ultima immagine, tale struttura è composta da:

- Un Network (TrueTime Network), rappresentante il creatore della rete 2. Infatti, ha i primi tre parametri impostati nella seguente maniera: “Network type” coincidente con il protocollo CSMA/CD (Ethernet), “Network number” pari a 2, che stabilisce che la rete realizzata è la numero 2, e “Number of nodes” posto uguale a 4, per sancire che i nodi di tale rete sono 4. Non ha né ingressi né uscite, ma mostra la porta di scheduling (Schedule output port);
- Un osservatore di risultati (Scope), che permette di visualizzare lo scheduling del blocco “TrueTime Network”.

I **controllori** implementati sono tre e si occupano di generare e mostrare il livello di infrazione, la multa, il segnale ed i punti relativi a tale infrazione e la segnaletica associata alla stessa. La loro struttura è la seguente:



Come si evince dalla prima delle ultime tre immagini, il **controllore dell'infrazione** è costituito da:

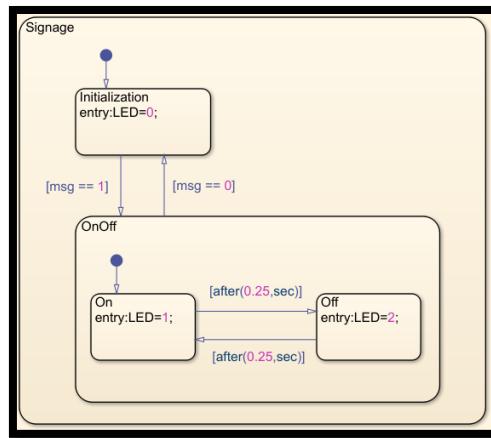
- Un kernel (TrueTime Kernel), rappresentante il nodo 2 della rete 2. Il kernel ha un ingresso e tre uscite, rappresentate dai dati di velocità, peso e cilindrata acquisiti dai sensori. Del kernel è mostrata anche la porta di scheduling (Schedule output port). La funzione passatagli è la “infringementController_init”;
- Un blocco di controllore a logica fuzzy (Fuzzy Logic Controller). Questo riceve in ingresso il risultato i dati di velocità, peso e cilindrata provenienti dal kernel, mentre la sua uscita, che corrisponde al livello di infrazione commessa dall’automobilista, viene trasferita al kernel stesso, per diventare l’ingresso. Al controllore a logica fuzzy viene passato il file “controller.fis”;
- Due osservatori di risultati (Scope), che permettono di visualizzare lo scheduling ed il livello di infrazione prodotto dal controllore a logica fuzzy.

Come si nota dalla seconda delle ultime tre immagini, il **controllore della multa** è costituito da:

- Un kernel (TrueTime Kernel), rappresentante il nodo 3 della rete 2. Il kernel ha tre ingressi ed un’uscita, rappresentata dal livello di infrazione prodotto dal precedente controllore. Del kernel è mostrata anche la porta di scheduling (Schedule output port). La funzione passatagli è la “fineFlashPointsController_init”;
- Una funzione MATLAB (MATLAB function). Questa riceve in ingresso il livello di infrazione proveniente dal kernel, mentre le sue uscite sono le informazioni sulla multa, sul segnale e sui punti della patente da mostrare all’automobilista, il cui raggruppamento viene trasferito al kernel stesso, per diventare l’ingresso;
- Un blocco di modulazione (Mux), che permette di usare un’unica connessione per trasferire tre informazioni;
- Un osservatore di risultati (Scope), che permette di visualizzare lo scheduling.

Come si osserva dalla terza delle ultime tre immagini, il **controllore della segnaletica** è costituito da:

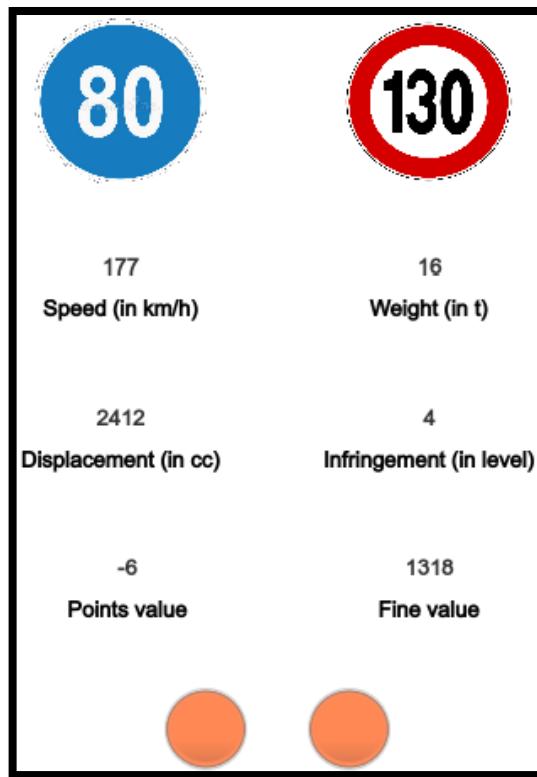
- Un kernel (TrueTime Kernel), rappresentante il nodo 4 della rete 2. Il kernel non ha un ingresso, ma possiede tre uscite. Di questo è mostrata anche la porta di scheduling (Schedule output port). La funzione passatagli è la “signageController_init”;
- Un grafico (Chart). Questo riceve in ingresso le informazioni sulla segnaletica da far notare all’automobilista e stabilisce quando quest’ultimo deve essere verde fisso o giallo lampeggiante. La sua struttura è la seguente:



Come si vede da quest’ultima immagine, la segnaletica viene inizialmente posta allo stato 0, vale a dire quello in cui è verde fisso. Qualora il livello di infrazione dovesse essere diverso da 0, la segnaletica viene posta per 0,25 secondi allo stato 1, quello in cui è giallo lampeggiante, e per i successivi 0,25 secondi nello stato 2, quello in cui è spento. Nel caso in cui, invece, il livello di infrazione dovesse risultare pari a 0, quindi nullo, la segnaletica rimane allo stato 0, ossia resta verde fisso.

- Un blocco di demodulazione (Demux), che divide i dati trasferiti con un’unica connessione in tre parti;
- Cinque osservatori di risultati (Scope), che permettono di visualizzare lo scheduling, la multa, la segnaletica che l’automobilista nota, il numero di punti della patente e gli stati della segnaletica.

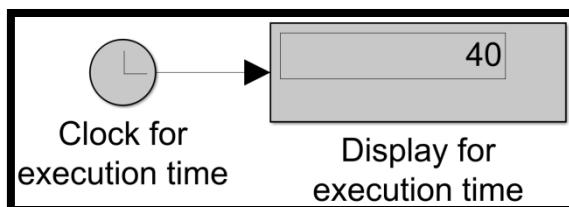
L’**interfaccia grafica** contenente tutte le informazioni rilevate e generate è la seguente:



In tale interfaccia, si possono visualizzare:

- Il segnale stradale di limite minimo da rispettare in autostrada di 80 km/h;
- Il segnale stradale di limite massimo da rispettare in autostrada di 130 km/h;
- La velocità del veicolo, espressa in chilometri orari, rilevata dal relativo sensore e mostrata sul display denominato “Speed (in km/h)”;
- Il peso del veicolo, espresso in tonnellate, rilevato dal proprio sensore e mostrata sul display denominato “Weight (in t)”;
- La cilindrata del veicolo, espressa in centimetri cubici, rilevata dal sensore associato e mostrata sul display denominato “Displacement (in cc)”;
- L’infrazione commessa dal conducente del veicolo, espressa in livelli da 0 a 5, ricavata dal controllore del livello di infrazione e mostrata sul display denominato “Infringement (in level)”;
- Il numero di punti da sottrarre alla patente del conducente del veicolo, ricavato dalla funzione MATLAB associata al controllore di multa, segnale e punti, e mostrato sul display denominato “Points value”;
- Il valore della multa da pagare da parte del conducente del veicolo, ricavato, anch’esso, dalla funzione MATLAB associata al controllore di multa, segnale e punti, e mostrato sul display denominato “Fine value”;
- Due segnalatori (Lamp), che possono mostrare una luce verde fissa o gialla lampeggiante in dipendenza dal valore del livello di infrazione. Il valore associato alla luce da mostrare è ricavato dal blocco “Chart” in cui è stato applicato il MBSD, il quale considera in input il segnale “Flash” ricavato dalla funzione MATLAB associata al controllore di multa, segnale e punti, e passato uguale al controllore di nome “signage controller”.

Nel grafico complessivo, è presente pure un blocco “Clock” abbinato ad un “Display” per la visualizzazione del tempo di esecuzione del sistema, come è possibile notare dalla seguente immagine:



V. VALUTAZIONE DELLE PERFORMANCE

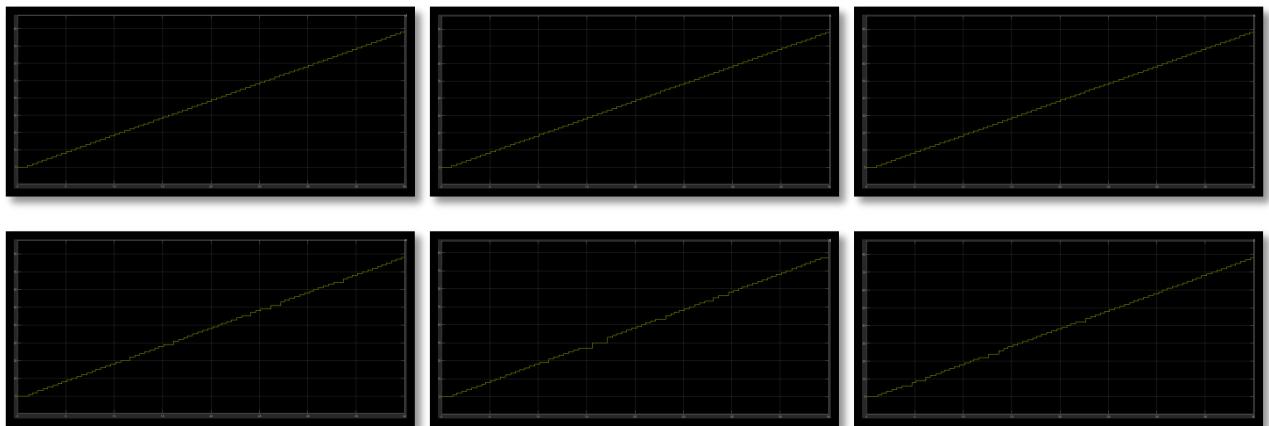
L'aspetto prestazionale della comunicazione in rete non è trascurabile al fine di migliorare l'intero sistema costruito. Per fare ciò, si effettuano rilevazioni ed, eventualmente, interventi mirati che contribuiscono al miglioramento dell'apparato nel suo complesso. In tale sistema, l'attenzione è posta principalmente sui tre nodi wireless presenti nel sistema, considerato che la rete cablata è priva o quasi di problemi legati al deterioramento in generale dell'informazione.

Le performance che si sono misurate sono relative al ritardo di propagazione dell'informazione (Delay) ed alla quantità di informazioni perse nell'etere (Packet loss).

Al passaggio di un veicolo, vengono effettuate le misurazioni da parte dei sensori e le informazioni vengono inoltrate di conseguenza. La tecnica di misurazione adottata è quella della previsione del valore numerico associato all'informazione successiva che si pensa di dover ricevere. Più semplicemente, ogni informazione possiede sia il dato misurato sia un numero identificativo univoco ed esso risulta essere incrementato rispetto all'informazione inoltrata precedentemente. In una situazione di questo tipo, la controparte si aspetta di ricevere tali dati in successione, ma non sempre accade ciò, in quanto qualche dato potrebbe andar perso nell'etere e di conseguenza si riceverà un valore che sicuramente sarà diverso, ed, in particolare, successivo a quello sperato.

Il calcolo delle informazioni perse viene effettuato ogni qualvolta si ha un valore incrementale diverso da quello sperato e, materialmente, mediante una sottrazione tra l'identificativo ricevuto e l'identificativo sperato. Questo risultato viene memorizzato insieme ai risultati precedenti della stessa natura.

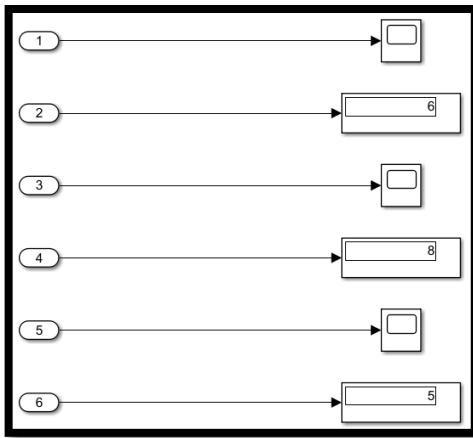
Di seguito, sono rappresentate, a coppie, le misurazioni condotte, rispettivamente, sui sensori di velocità, peso e cilindrata:



Nelle tre immagini della riga di sopra si evince il quantitativo informativo sulla velocità, sul peso e sulla cilindrata inviato in 40 secondi. Le informazioni vengono inviate con una frequenza di rilevazione di mezzo secondo, a partire dall'istante di 1 secondo della simulazione, perciò si hanno 78 inoltri. Nelle immagini della riga di sotto si notano istanti temporali di non pervenuta informazione, vale a dire le linee piatte. In effetti, non si ha un incremento regolare in base al tempo come accadeva per gli inoltri effettuati nelle immagini della riga di sopra. Si stima che per tutta la durata della simulazione, vale a dire nei 40 secondi previsti, si ha:

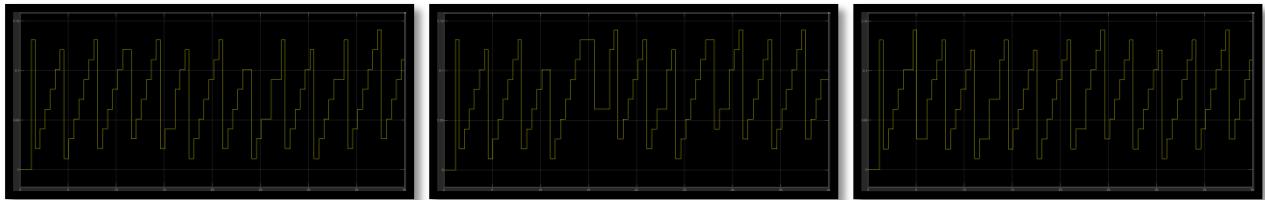
- Una perdita di 6 pacchetti sulla velocità, ovvero 72 ricezioni su 78 inoltri, con una precisione del 92,31%;
- Una perdita di 8 pacchetti sul peso, vale a dire 70 ricezioni su 78 inoltri, con una precisione del 89,74%;
- Una perdita di 5 pacchetti sulla cilindrata, ossia 73 ricezioni su 78 inoltri, con una precisione del 93,59%.

Nella successiva immagine, le ricezioni dei pacchetti di velocità, peso e cilindrata appena viste sono visualizzabili, rispettivamente, negli osservatori di risultati, vale a dire negli scope, collegati alle porte 1-3-5. Invece, i display connessi con le porte 2-4-6 mostrano il numero di pacchetti persi per ognuna delle precedenti informazioni, nello stesso ordine.

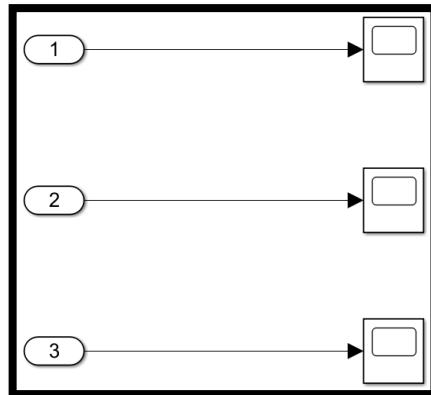


Il ritardo di propagazione è il tempo intercorso dalla trasmissione dell'informazione fino al tempo di avvenuta ricezione. Questa rilevazione è fondamentale per evitare se si vengono a verificare periodi temporali in cui l'informazione subisce fenomeni di "Multipath fading" causati dall'ambiente circostante. Questo è un problema relativo alle comunicazioni wireless, in cui il segnale può subire una variazione, dovuta ad altri sistemi che emettono segnali elettromagnetici sulla stessa lunghezza d'onda o su una molto simile. Il fenomeno è più calzante se si discute di deterioramento dell'informazione e non di delay. Più concretamente, il ritardo di propagazione è determinato dagli ostacoli che il segnale incontra lungo la propagazione nell'etere, quali edifici, avverse condizioni meteorologiche o utilizzo del medesimo canale di trasmissione per apparati wireless analoghi. In ambiente simulato, si è posti l'esigenza di creare un ambiente quanto più disturbato possibile, impostando opportuni criteri, al fine di testare il sistema in condizioni più o meno sfavorevoli alla normale comunicazione priva di ogni disturbo, realizzando quello che viene definito come stress test.

I ritardi di propagazione misurati nelle prossime tre immagini sono relativi ai sensori di velocità, peso e cilindrata. Tali ritardi sono tra loro molto simili. Da queste immagini si nota come ad ogni inoltro di mezzo secondo corrisponde mediamente nell'arco dei 40 secondi un ritardo medio che si attesta intorno all'8%, corrispondente al valore 0,08 dell'asse verticale nelle immagini.



Nella successiva immagine, i ritardi di propagazione riguardanti i sensori di velocità, peso e cilindrata appena visti sono visualizzabili, rispettivamente, negli osservatori di risultati, vale a dire negli scope, collegati alle porte 1-2-3.



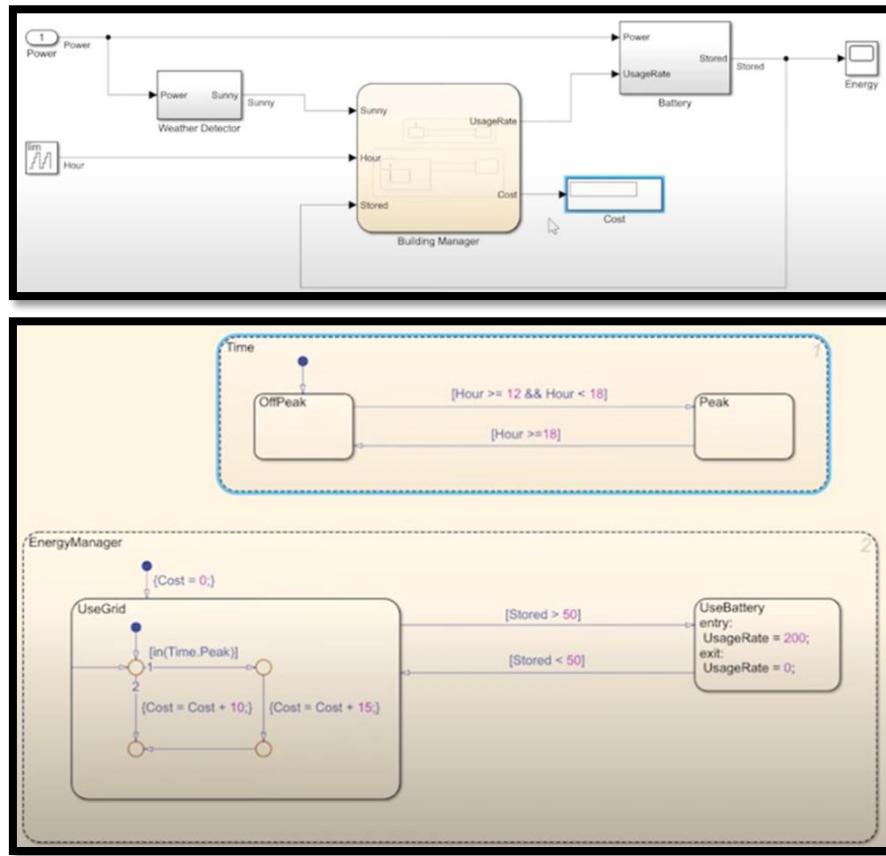
VI. CONCLUSIONI

Il sistema DVA proposto si integra perfettamente con la rete autostradale italiana e nulla esclude di poterlo implementare in altri paesi, in cui il tasso di incidenti o situazioni pericolose in autostrada è elevato.

Già in altre reti stradali, sistemi di questo tipo sono installati con una certa frequenza e hanno contribuito a ridurre questi tassi. Nelle autostrade, invece, tali sistemi non sono presenti in grande quantità, se non in quelle più importanti e trafficate. Infatti, le autostrade meno percorse dagli automobilisti sono parzialmente o totalmente prive di questi ed altri sistemi di prevenzione di circostanze dannose. In ogni caso, nella maggior parte delle autostrade italiane sono presenti gli autovelox, ma non sempre svolgono il proprio compito, mentre, quando agiscono concretamente e gli automobilisti ricevono delle multe, capita spesso di venire a conoscenza di situazioni in cui questi presentano ricorsi, in quanto, a detta loro, gli autovelox sembrano non funzionanti. Oltre a questo, gli automobilisti sono costretti a dover attendere i tempi della burocrazia per conoscere l'esito dei ricorsi presentati e tutto ciò causa in loro malumore e nervosismo, che potrebbe ripercuotersi nelle prossime guide, col rischio di essere meno lucidi ed attenti e provocare eventi dannosi.

Ad ovviare a questi problemi si occupa il sistema DVA, che ha semplicemente il compito di avvertire gli automobilisti della velocità che stanno mantenendo e di mostrare le apposite sanzioni previste per i casi in cui essa non risiede tra il limite minimo ed il limite massimo, senza generarle materialmente.

Il sistema potrebbe essere oggetto di futuri sviluppi. Infatti, è possibile integrare un pannello fotovoltaico governato tramite logica stateflow opportuna, che riesca a gestire la carica della batteria a fronte dell'uso dell'energia elettrica. La logica stateflow potenzialmente implementabile è più vicina a quella rappresentata dall'esempio mostrato nella seguente immagine e proposto dalla stessa azienda "Matworks", proprietaria del programma MATLAB.



Il sistema DVA è un sistema scalabile e, anche se nella trattazione si fa riferimento al servizio in autostrada, non si hanno limitazioni in termini di utilizzo di tale sistema anche in altre reti stradali, quali le strade extraurbane, sia principali sia secondarie, le strade urbane, sia di scorrimento sia di quartiere, e le strade locali. In tali ambienti, il sistema andrebbe aggiornato opportunamente sui limiti minimi e massimi, sui livelli di infrazione, sul quantitativo della multa e sul numero dei punti della patente.

BIBLIOGRAFIA

- 1) Reti di calcolatori e Internet. Un approccio top-down; 7° edizione; James F. Kurose, Keith W. Ross, ISBN 13: 9788891902542
- 2) Reti di calcolatori; 5° edizione; Andrew S. Tanenbaum, David J. Wetherall; ISBN 13: 9788891908254
- 3) Essential MATLAB for Engineers and Scientists; 5° edizione; Brian H. Hahn, Daniel T. Valentine; ISBN 13: 9780123943989
- 4) Numerical Analysis Using MATLAB and Excel; 3° edizione; Steven T. Karris; ISBN 13: 9781934404041
- 5) Matlab Simulink documentation MathWorks: <https://it.mathworks.com/help/matlab/>
- 6) TrueTime 2.0 – Reference Manual; Anton Cervin, Dan Henriksson, Martin Ohlin; Department of Automatic Control Lund University; February 2016
- 7) Elenco funzioni TrueTime: <https://github.com/sfischme/TrueTime/tree/master/kernel/matlab/help>
- 8) <https://www.wroar.net/pages/dissuasori-di-velocita.html>
- 9) <https://www.focus.it/tecnologia/innovazione/tutor-autovelox-e-gli-altri-come-funzionano>
- 10) <https://www.quotidianomotori.com/automobili/veri-limiti-velocita-strade-autostrade>
- 11) <https://www.brocaldi.it/codice-della-strada/titolo-i/art2.html>