

Alma Mater Studiorum – Università di Bologna

DOTTORATO DI RICERCA IN
Meccanica e scienze avanzate dell'ingegneria

Ciclo XXXI

Settore Concorsuale: 09/C1

Settore Scientifico Disciplinare: ING-IND/09

SVILUPPO DI SISTEMI PER L'AUTOMAZIONE DEI TEST SUI MOTORI

Presentata da: Ing. Michele Taccioli

Coordinatore Dottorato

Supervisore

Prof. Marco Carricato

Prof. Enrico Corti

Esame finale anno 2019

Con la scrittura di questo elaborato si conclude una fase importante della mia vita: tre anni di lavoro intenso, sacrifici e anche soddisfazioni, un miscuglio di esperienze che hanno sicuramente portato ad una crescita personale.

Tutte le gioie e le cicatrici che mi hanno segnato durante questo cammino svanirebbero immediatamente nel nulla senza averle condivise con un amico con cui ricordarle un giorno ridendo davanti ad un buon bicchiere di vino. Aldilà di aver vissuto un'esperienza incredibile, la mia vera fortuna è stata quella di aver lavorato con tante persone in gamba che non mi hanno mai fatto mancare il loro sostegno, spronandomi a fare sempre meglio.

Credo che il primo ringraziamento vada indubbiamente al mio compagno di sventure Filippo Carra che mi ha accompagnato in ogni fase di questo percorso, riuscendo nel difficile compito di stemperare le ansie e le preoccupazioni giornaliere con due cazzate e una birra, una volta usciti dall'hangar.

Ringrazio Federica che sebbene impegnata e spesso lontana da casa per inseguire giustamente, come me, le proprie aspirazioni, non mi ha mai fatto mancare il suo affetto, anche quando entrambi non riuscivamo a parlare al telefono per la stanchezza.

Ringrazio il Prof. Corti per avermi offerto questa opportunità, riponendo in me la sua fiducia. Ringrazio anche il Prof. Ponti, tutti i soci e i ragazzi di Alma, un gruppo di cui è bello far parte.

Ringrazio i ragazzi dello sviluppo del Reparto Corse di Ducati, che mi accolgono sempre con tanto calore e concedendomi di lavorare per loro mi regalano un briciolo di quel mondo che ci appassiona tanto.

Ringrazio la mia famiglia, che non ha mai digerito questa mia scelta, non la capisce e probabilmente non la capirà mai. Non fa nulla, la loro testardaggine mi spinge a rimanere più fermamente sulla mia strada, spinto più dalla passione, che da altri obiettivi.

Ringrazio anche quei cretini di Alfero che sono con me da una vita, gli anni passano, succedono cose, ma noi, fortunatamente, non ci perdiamo mai. Grazie, perché quando poi si fa serata insieme mi ricordate che va bene lavorare duro ma poi ci si DEVE divertire forte.

Infine ringrazio la comunità del hangar: porterò sempre con me il ricordo delle tante emozioni vissute in questo luogo per me magico.

Abstract

These days powertrains development has reached such a high level of complexity in terms of mechanical solutions and electronic components development that the “calibration effort”, time spent on engine testing to validate solutions and optimize the engine management highly increased. In fact, each device installed on-board requires a fine tuning to be properly integrated in the engine control system. So, the role of the test bench has become essential in order to validate both mechanical solutions and engine control strategies introduced during the development stage.

The PhD research was focused on the development of methodologies for automated testing on internal combustion engines. More specifically, starting from the previous application dedicated to the UNIBO test benches control, a new software, based on a Real-Time compatible platform, named NI Veristand, has been implemented. Once created a project with all the basic functionalities for the engine testing, it has been significantly expanded developing and integrating custom tools and models to add new features. A relevant part of the activity has been devoted to the bench conditioning systems (engine oil, coolant and intake air conditions) management and the development of bench automations. Control models and automations have been validated performing several types of tests on Unibo and Alma-Automotive’s engine test benches.

Sommario

1. Sala prove motore.....	6
2. La gestione del banco.....	13
2.1 Requisiti	13
2.2 Layout hardware.....	15
2.3 Ambiente di sviluppo software: NI Veristand	18
2.3.1 NI Veristand Engine	20
2.3.2 System Explorer	24
2.3.3 Workspace e UI Manager	33
3. Sviluppo di oggetti custom per NI Veristand.....	36
3.1 Fpga Personality.....	36
3.2 Models	39
3.3 Custom Devices.....	42
4. Il controllo dei sistemi di condizionamento del banco	46
4.1 Plant dei sistemi di condizionamento.....	47
4.1.1 Layout unità di trattamento dell'aria comburente	47
4.1.2 Layout sistema di condizionamento dell'acqua motore	50
4.1.3 Layout sistema di condizionamento dell'olio motore.....	51
4.2 Algoritmi di controllo dei sistemi di condizionamento	55
4.2.1 Controllo condizioni aria comburente	56
4.2.2 Controllo temperatura acqua motore.....	64
4.2.3 Controllo temperatura olio motore	67
4.3 Risultati	71
4.3.1 Calibrazione del controllo.....	72
4.3.2 Validazione del controllo	74

5. Modello termico del sistema di raffreddamento motore	79
5.1 Schema dell'impianto.....	79
5.2 Modello termico motore	81
5.3 Modello termico radiatore	82
5.4 Calcolo delle portate	84
5.5 Risultati delle simulazioni.....	87
6. L'automazione della sala prove	91
6.1 Le sequenze Real-Time	93
6.2 Test Manager tool.....	96
6.2.1 Struttura	97
6.2.2 Interfaccia	99
6.3 La calibrazione anticipo	105
6.3.1 Calibrazione classica	106
6.3.2 Calibrazione EVO	108
7. Conclusioni.....	112
Definizioni e abbreviazioni.....	115
Riferimenti bibliografici	121

Indice delle figure

<i>Figura 1-1: Sala prove motore AMG</i>	6
<i>Figura 1-2: Layout dei sistemi presenti in sala prove</i>	7
<i>Figura 1-3: Rassegna di sensori comunemente utilizzati in sala prove</i>	9
<i>Figura 1-4 Centralina di controllo motore con relativo software di calibrazione</i>	12
<i>Figura 2-1: Architettura sistema real-time embedded</i>	16
<i>Figura 2-2: Piattaforma hardware del sistema di gestione sala prove</i>	18
<i>Figura 2-3: Layout di Veristand</i>	20
<i>Figura 2-4: Struttura NI Veristand Engine</i>	21
<i>Figura 2-5: System Explorer</i>	24
<i>Figura 2-6: Hardware</i>	26
<i>Figura 2-7: Custom Devices</i>	27
<i>Figura 2-8: Models</i>	28
<i>Figura 2-9: Alarms</i>	29
<i>Figura 2-10: Procedures</i>	30
<i>Figura 2-11: Aliases</i>	31
<i>Figura 2-12: Scales</i>	32
<i>Figura 2-13: System Mappings</i>	33
<i>Figura 2-14: Workspace</i>	35
<i>Figura 2-15: UI Manager</i>	35
<i>Figura 3-1: Acquisizione di due moduli NI9215</i>	37
<i>Figura 3-2: Acquisizione modulo NI 9375</i>	37
<i>Figura 3-3: FPGA Personality: Communication Loop</i>	38
<i>Figura 3-4: Esempio di configurazione di un segnale in fpgaconfig</i>	39
<i>Figura 3-5: Esempio di modello implementato in NI Labview: controllo PID</i>	41
<i>Figura 3-6: Esempio di modello implementato in Simulink: gestione dinamiche delle soglie di allarme</i>	42
<i>Figura 3-7: Struttura di un Custom Device</i>	45

<i>Figura 4-1: Layout unità di trattamento aria.....</i>	<i>48</i>
<i>Figura 4-2: Layout sistema di condizionamento acqua e olio motore</i>	<i>52</i>
<i>Figura 4-3: Pre-riscaldamento e umidificazione</i>	<i>57</i>
<i>Figura 4-4: Pre-riscaldamento, umidificazione e post-riscaldamento</i>	<i>58</i>
<i>Figura 4-5: Deumidificazione e post-riscaldamento.....</i>	<i>59</i>
<i>Figura 4-6: Raffreddamento e umidificazione</i>	<i>60</i>
<i>Figura 4-7: Diagramma a blocchi controllo UTA.....</i>	<i>64</i>
<i>Figura 4-8:Diagramma a blocchi controllo temperatura acqua motore.....</i>	<i>67</i>
<i>Figura 4-9:Diagramma a blocchi controllo temperatura olio motore</i>	<i>71</i>
<i>Figura 4-10: Regime motore, carico e flusso d'aria comburente durante una curva di potenza</i>	<i>72</i>
<i>Figura 4-11: Variazioni a gradino della temperatura aria target per la taratura del PI.....</i>	<i>73</i>
<i>Figura 4-12: Variazioni a gradino dell'umidità relativa per la taratura del PI.....</i>	<i>74</i>
<i>Figura 4-13: Controllo temperatura e umidità aria comburente durante una curva di potenza</i>	<i>75</i>
<i>Figura 4-14: Controllo temperatura acqua motore durante una curva di potenza</i>	<i>77</i>
<i>Figura 4-15: Controllo temperatura olio motore durante una curva di potenza.....</i>	<i>77</i>
<i>Figura 4-16: Controllo acqua e olio motore durante una simulazione di giro pista</i>	<i>78</i>
<i>Figura 5-1: Schema del sistema di raffreddamento.....</i>	<i>80</i>
<i>Figura 5-2:Potenza ceduta dal radiatore all'aria, confronto tra dati sperimentali e risultati del modello.....</i>	<i>83</i>
<i>Figura 5-3: Analogia elettrica del circuito di raffreddamento</i>	<i>84</i>
<i>Figura 5-4: Caratteristiche delle due resistenze elettriche equivalenti alla valvola termostatica.....</i>	<i>85</i>
<i>Figura 5-5: Test di riscaldamento con motore al minimo.....</i>	<i>87</i>
<i>Figura 5-6: Test con veicolo ad alta velocità.....</i>	<i>88</i>
<i>Figura 5-7: Condizioni giro pista normalizzate</i>	<i>89</i>
<i>Figura 5-8: Simulazione giro pista</i>	<i>90</i>
<i>Figura 5-9: (a) Confronto portate, (b) Confronto prevalenze, (c) Confronto temperature refrigerante.....</i>	<i>90</i>
<i>Figura 6-1: Una sequenza configurata su NI Stimulus Profile editor</i>	<i>95</i>
<i>Figura 6-2: Costruzione della sequenza per la simulazione del giro pista.....</i>	<i>96</i>

<i>Figura 6-3: Struttura della macchina a stati queue driven con producer e consumer loop</i>	99
<i>Figura 6-4: TMan – Home</i>	100
<i>Figura 6-5: TMan: Cycle Settings</i>	101
<i>Figura 6-6: TMan: Step Settings</i>	102
<i>Figura 6-7: TMan - Tx Check</i>	103
<i>Figura 6-8: TMan - Check before step</i>	104
<i>Figura 6-9: TMan - Step execution</i>	105
<i>Figura 6-10: Calibrazione anticipo classica</i>	107
<i>Figura 6-11: Interfaccia algoritmo di calibrazione</i>	111

Introduzione

Il motore a combustione interna vive in un costante stato di frenetica evoluzione per cercare di adeguarsi ad un mondo sempre attento alle tematiche ambientali. Il continuo inasprimento delle norme antinquinamento comporta enormi sforzi per i costruttori nello studio di soluzioni sempre più complesse. Tante sono le case automobilistiche che già qualche anno fa, specialmente dopo il ben noto scandalo del “dieselgate”, hanno preso posizione schierandosi dalla parte dei motori elettrici, abbandonando lo sviluppo di propulsori diesel per spingersi invece sull’ibridizzazione dei propri veicoli e ipotizzando la scomparsa dell’ormai obsoleto motore a combustione interna nel giro di pochi decenni. Previsioni più o meno fantasiose a parte, il motore endotermico accompagna la storia e lo sviluppo dell’umanità da più di un secolo e come tutto ciò che ci circonda, da noi creato, deve evolvere anche esso per rispondere alle nostre attuali esigenze.

Al giorno d’oggi, l’aspetto “green”, ovvero basse emissioni e consumi ridotti, è un requisito fondamentale per un motore a combustione interna, ma non è il solo. Infatti, anno dopo anno, i costruttori si sono imposti standard di affidabilità e prestazioni sempre più elevati per proporre ai propri clienti un prodotto in grado di offrire un’esaltante e confortevole esperienza di guida e, allo stesso tempo, interventi di manutenzione sempre meno frequenti.

Per ottenere un motore endotermico in grado di coniugare tutti questi aspetti è necessario un meticoloso lavoro di progettazione e sviluppo focalizzato sullo studio di soluzioni tecniche complesse ma al contempo affidabili e l’integrazione massiccia di componenti elettronici. Questi ultimi, in particolare, amplificano notevolmente le funzionalità del motore endotermico, permettendo al propulsore di adeguare le proprie prestazioni sulla base delle richieste del pilota, delle condizioni dell’ambiente ad esso circostante, dello stato dei componenti in esso installati, ecc.. I soli organi meccanici non consentirebbero al motore di adattarsi e spaziare un così vasto campo di funzionamento. Dunque, per rompere i rigidi schemi imposti dalla meccanica “fredda”, gli ingegneri ricorrono costantemente all’elettronica dando vita ad un sistema con una propria intelligenza, l’unità di controllo motore, che monitora, analizza e infine decide come controllare i vari attuatori che equipaggiano il propulsore per far fronte nella maniera più opportuna alle esigenze sopra citate.

Tuttavia, ogni dispositivo elettronico che viene installato a bordo comporta un aggravio del cosiddetto “calibration effort”, ovvero l’impegno dedicato alla fase di test e calibrazione per far sì che il componente venga correttamente integrato nel sistema di controllo motore. Maggiore è il numero di attuatori controllati, maggiore è il numero di leve di controllo che si hanno a disposizione per variare il comportamento del motore e di conseguenza maggiore sarà il numero di prove sperimentali da svolgere al banco prova, ancora prima che il propulsore possa essere installato in vettura.

Dunque, il ruolo della sala prove motore è divenuto essenziale [1], in qualità di strumento che consente di svolgere i test più disparati, collezionando una considerevole mole di dati. L’esperienza che si acquisisce in sala prove permette non solo di ottimizzare con opportune calibrazioni il controllo motore ma anche di validare soluzioni tecniche ed eventualmente evidenziare punti critici che, in questa fase preliminare, possono ancora essere risolti con un minor costo rispetto ad una fase più matura quale ad esempio i test in vettura.

All’interno di una sala prove motore sono presenti alcune attrezzature comuni ad ogni laboratorio in cui si desidera effettuare attività di testing su motore a combustione interna [1, 2, 3]. Oltre a queste attrezzature “standard”, l’equipaggiamento della test room può variare sulla base non solo del tipo di prova da effettuare (durata, calibrazione, curve di potenza, avviamento, shock termico, funzionamento con forti inclinazioni, ecc.) ma anche del tipo di motore oggetto dei test. Solitamente, tali attrezzature sono progettate per svolgere ognuna una gamma di funzioni molto specifica e limitata, e l’eterogeneità di tali componenti fa sì che l’equipaggiamento della sala prove non si presti ad una gestione di tipo globale. Tuttavia, l’aspetto della comunicazione tra e con i vari devices del banco prova è essenziale per sfruttare al meglio tutte le informazioni che vengono generate durante i test. È dunque necessario che il sistema di gestione della sala includa, all’interno della propria logica, l’implementazione dei protocolli di comunicazione digitale standard che permettono la condivisione di dati fra i vari componenti.

Altro aspetto, non meno importante, è l’automazione della sala prove: per raggiungere un buon livello di accuratezza e ripetibilità, velocizzando al tempo stesso lo svolgimento delle prove, diviene fondamentale limitare, per quanto possibile, l’intervento manuale dell’operatore di cella durante lo svolgimento dei test. Inoltre, soprattutto in fase di

calibrazione, il numero di parametri da gestire o monitorare è oramai talmente elevato che non è più umanamente possibile un approccio manuale di gestione della prova.

In questo contesto si pone l'attività di dottorato di questi tre anni: lo sviluppo di sistemi per l'automazione dei test sui motori a combustione interna. Attraverso la collaborazione con Alma-Automotive S.r.l e il Reparto Corse di Ducati Motor Holding S.p.A. si è provveduto allo sviluppo di un sistema di controllo per sala prove, a basso costo di gestione e sviluppo, compatibile con piattaforma hardware Real-Time prodotta da National Instrument, che include, oltre alle funzionalità base per svolgere attività di testing sui motori a combustione interna (data logging, allarmi, procedure di recovery, canali matematici, ecc.), logiche model-based per il controllo di alcuni strumenti del banco prove. Il sistema di controllo agisce come un collettore di informazioni potendo dialogare con i vari devices, grazie all'implementazione di alcuni dei più comuni protocolli di comunicazione oggi utilizzati in questo campo. Per quanto riguarda l'automazione della sala prove, una parte dell'attività di dottorato ha riguardato lo sviluppo di un tool che permette all'operatore di configurare e lanciare delle sequenze precompilate tramite cui è possibile interagire con la centralina per variare i parametri di controllo motore: un requisito fondamentale per le attività di sperimentazione e calibrazione.

L'attività svolta durante il periodo di dottorato viene presentata in questo elaborato articolandosi nel seguente modo:

- Il primo capitolo è una breve introduzione alla sala prove motore, una panoramica di quello che è l'equipaggiamento necessario per svolgere attività di testing sui motori a combustione interna.
- Il secondo capitolo descrive il sistema di gestione della sala prove, presentando la piattaforma hardware designata per rispondere alle esigenze del controllo e riportando le motivazioni della scelta di NI Veristand come ambiente di sviluppo per il software di controllo del banco prove.
- Il terzo capitolo illustra alcuni esempi di oggetti implementati e compilati in un ambiente di sviluppo diverso da NI Veristand per aggiungere funzionalità non

nativamente presenti nella piattaforma software scelta per sviluppare il software della sala prove.

- Il quarto capitolo propone un approccio di controllo model-based e centralizzato, sviluppato per la gestione dei sistemi di sala per il condizionamento di temperatura acqua motore, temperatura olio motore, umidità e temperatura aria in aspirazione motore.
- Il quinto capitolo presenta lo sviluppo di un modello termico del sistema di raffreddamento di un motore: questo argomento, apparentemente svincolato dal tema principale di questo elaborato, è stato affrontato per diverse motivazioni, tra le quali anche quella di migliorare la gestione dei sistemi di condizionamento dei fluidi motore in sala prove.
- Il sesto capitolo descrive l'applicazione creata per la configurazione e la gestione dei cicli automatici utilizzati durante le attività di sperimentazione sul propulsore e calibrazione dei vari parametri di controllo motore.

1. Sala prove motore

La complessità e l'equipaggiamento di un banco prove sono strettamente legati sia alla natura dei test che devono essere svolti sia alla tipologia del motore oggetto delle prove. Nonostante le particolarità e i componenti specifici che possono rendere una sala prove diversa da un'altra, un banco per fare sperimentazione sui motori, in generale, può essere schematizzato come un insieme di componenti che operano coordinatamente, gestiti e monitorati da un'unità centralizzata, per permettere lo svolgimento di prove di vario genere, acquisendo dati ed esperienze utili per lo sviluppo del propulsore e del suo sistema di controllo.

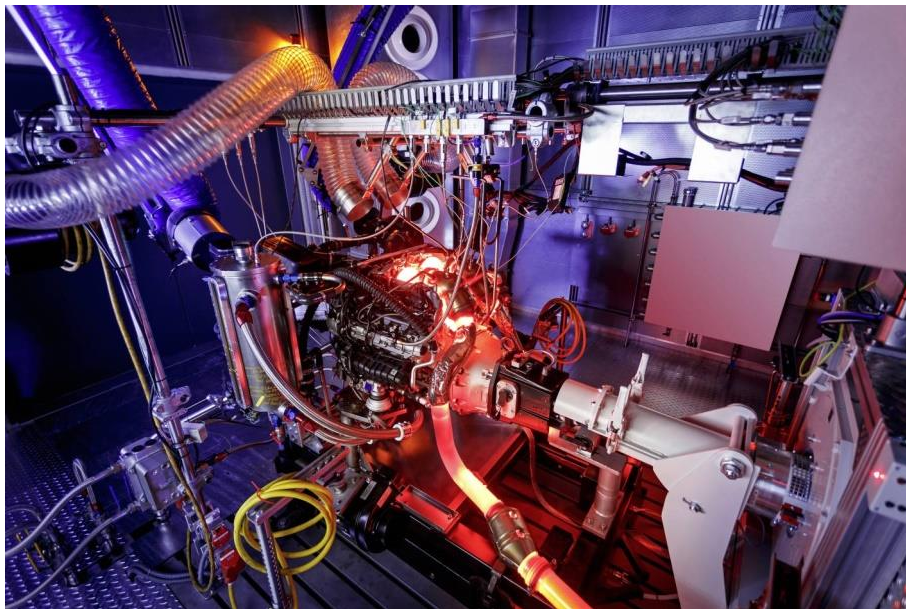


Figura 1-1: Sala prove motore AMG

Nello specifico, l'hardware solitamente presente in una sala prove motore può essere suddiviso nei vari sistemi evidenziati nello schema presentato in Figura 1-2:

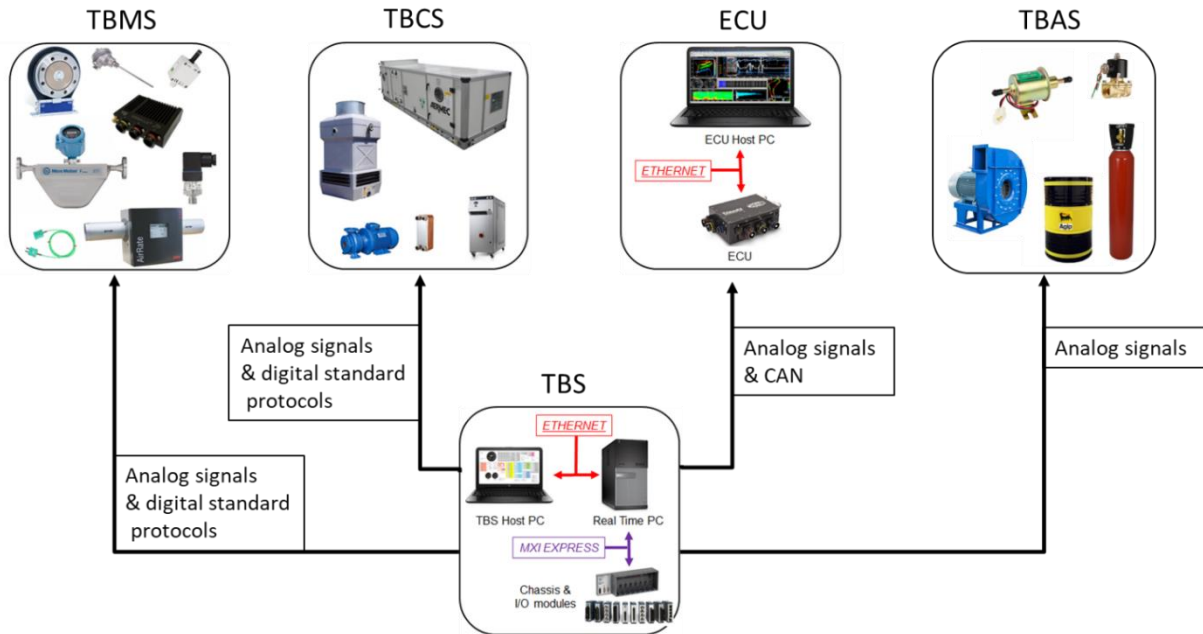


Figura 1-2: Layout dei sistemi presenti in sala prove

- **Test Bench Supervisor (TBS):** il supervisore è quel sistema che svolge il compito di gestire la sala prove, solitamente controllato dall'operatore di cella, ma talvolta, se opportunamente programmato, anche in totale autonomia. Esso interagisce con gli altri sistemi di cella al fine di eseguire i seguenti task:
 - Controllare il motore in termini di giri e carico (demand acceleratore), in modo da poterlo portare alle condizioni operative dettate dal test.
 - Acquisire a frequenza opportuna tutte le grandezze monitorate tramite i sensori presenti in cella. Il supervisore in quanto tale, agisce come collettore di tutte le informazioni prodotte durante il test e salva questi dati in file che devono essere compatibili con gli strumenti di analisi e post-trattamento utilizzati per leggere i risultati della prova.
 - Gestire con accuratezza le attuazioni di cella [4] (interazione con i sistemi di condizionamento) al fine di mantenere al valore target prestabilito i parametri

relativi, ad esempio, ad aria comburente, liquido refrigerante, lubrificante e carburante.

- Svolgere le azioni sopra elencate nella più totale sicurezza, in primis per gli operatori di cella, ma anche per i componenti del motore e delle attrezzature di cella. A tale scopo, quelli che sono ritenuti essere i segnali più critici, non solo vengono acquisiti ma anche allarmati. Nel caso in cui uno di tali parametri oltrepassi le soglie prestabilite, il supervisore, riscontrando l'allarme, deve applicare delle procedure di recovery (portare il motore in idle o spegnerlo più o meno velocemente in base alla tipologia di emergenza) e mettere il banco in sicurezza.
- Infine, il supervisore deve consentire all'operatore di cella di gestire e monitorare lo svolgimento delle prove tramite un'interfaccia, che deve essere il più possibile intuitiva e leggibile.

La piattaforma hardware del sistema supervisore sarà argomento del prossimo capitolo insieme alla descrizione dell'ambiente di sviluppo scelto per l'implementazione del software di gestione della sala prove.

- **Test Bench Measurement System (TBMS):** come mostra la Figura 1-3, si tratta di tutti i sensori e i devices che monitorano i parametri del motore e della cella. Le misure vengono trasmesse al supervisore tramite segnali analogici in tensione o in corrente. Sempre più strumenti prevedono inoltre la possibilità di trasmettere i dati in formato digitale, una comunicazione meno affetta da disturbi (problema di non poco conto in una sala prove). Al fine di poter sfruttare questa funzionalità è necessario che all'interno del software di gestione della sala prove siano implementati i protocolli di comunicazione digitale standard (TCP/IP, CAN, CanOpen, Modbus RS 485, Modbus RS 232, ecc).



Figura 1-3: Rassegna di sensori comunemente utilizzati in sala prove

- **Test Bench Conditioning Systems (TBCS):** sistemi necessari al controllo, durante i test, di tutti quei fattori, esterni al motore, che possono influenzare lo sviluppo della combustione e, più in generale, il risultato delle prove in termini di prestazioni, emissioni e affidabilità dei componenti del propulsore.

Tra i fattori da controllare sono sicuramente presenti le temperature del liquido refrigerante e dell'olio motore. Mentre nell'installazione sul veicolo questi parametri vengono gestiti sfruttando lo scambio termico che avviene all'interno dei radiatori, tra l'aria che impatta frontalmente il mezzo e i fluidi motore (il liquido refrigerante e l'olio), in sala prove si sfrutta, al posto dell'aria, il liquido fornito dal sistema di raffreddamento dei componenti del banco. Lo scambio termico tra tale liquido e i fluidi motore avviene all'interno di scambiatori di calore a piastre opportunamente dimensionati. Gestendo la portata di fluido cella in ingresso agli scambiatori è possibile regolare lo scambio termico al fine di mantenere le temperature dei fluidi motore in target. Inoltre, in alcuni casi, può essere prevista anche la possibilità di gestire l'incremento di temperatura tramite un circuito che include un riscaldatore elettrico di acqua o olio diatermico, che vengono inviati ad uno scambiatore in cui cedono calore al fluido motore da condizionare. Questa funzionalità risulta essere molto vantaggiosa in termini di tempo e combustibile risparmiato all'atto dell'accensione, consentendo di portare velocemente il motore in temperatura. Inoltre,

consente di svolgere test che richiedono condizioni particolari, come ad esempio olio motore caldo nonostante il motore sia in condizioni operative che lo portano ad erogare poca potenza.

In alcuni casi, ma non sempre, vengono controllate le condizioni (temperatura, pressione e umidità) dell'aria comburente. Solitamente, vista l'influenza che tali parametri hanno sulle prestazioni rilevate al banco prove, laddove questi fattori non siano gestiti tramite opportuno hardware, si applica banalmente un coefficiente correttivo che tiene conto del valore di tali parametri. Tuttavia, la correzione applicata risulta essere valida solo entro certi limiti. Inoltre, se lo scopo del test è proprio quello di verificare il comportamento e l'affidabilità del motore quando soggetto a condizioni estreme, non naturalmente riproducibili al banco, è necessario equipaggiare la sala prove con un'apposita unità di trattamento aria.

Infine, in alcuni casi viene previsto anche un sistema di condizionamento della temperatura del combustibile, al solo scopo di evitare che la parte di carburante che si è scaldata, poiché rifiutata dal regolatore di pressione, venga reimpressa nel circuito di alimentazione e inviato al motore ad una temperatura troppo elevata.

- **Test Bench Auxiliary Systems (TBAS)**, che include:
 - il sistema di ventilazione della cella, capace di effettuare il totale ricambio dell'aria (lavaggio) presente in sala, con una certa frequenza dettata da norme ben precise;
 - il sistema di alimentazione del combustibile, composto da fusti contenuti in appositi armadi o da serbatoi di stoccaggio, elettrovalvole di sicurezza sul circuito, pompe esterne e/o interne alla cella e sensore di portata combustibile (può essere di diverse tipologie: Coriolis, ultrasuoni, bilancia, ecc..)
 - il sistema antincendio, composto da centralina e sensori dedicati che in caso di incendio attivano, solo se confermata la chiusura delle porte di cella (che indica anche assenza di operatori in sala prove) e dopo aver disattivato le ventilazioni, il rilascio di CO₂.

- **Engine Control Unit System (ECU):** la centralina controllo motore è, insieme al motore stesso, l'oggetto della sperimentazione. In quanto tale, essa deve rendere disponibili le informazioni relative alle attuazioni che vengono richieste per gestire il propulsore durante la prova, oltre alle misure derivanti dai sensori che equipaggiano di serie il propulsore e che l'unità di controllo motore monitora per svolgere i suoi calcoli. Solitamente la ECU comunica le attuazioni, in tempo reale via TCP/IP, ad un software di calibrazione (fornito dallo stesso sviluppatore dell'unità di controllo motore) installato su di un host PC, che funge da interfaccia diretta tra l'operatore di cella e la centralina. Tuttavia, come già sottolineato in precedenza, il supervisore agisce come un collettore di dati e ha il compito di salvare all'interno dello stesso file di acquisizione i dati provenienti da tutti i devices presenti in sala. A tale scopo, se previsto dal software implementato in ECU, quest'ultima può pubblicare su bus CAN una ristretta cerchia di parametri strategici che vengono dunque acquisiti dal sistema di gestione del banco.

Poter monitorare le attuazioni applicate dalla centralina è di fondamentale importanza in fase di validazione del software implementato in essa. Tuttavia, per poter operare una corretta calibrazione dei parametri di controllo motore è necessario che il colloquio sia bidirezionale, ovvero che sia prevista in centralina la possibilità di sovrascrivere dall'esterno il valore di alcuni parametri o addirittura di ricevere nuove calibrazioni. Ovviamente ciò è possibile (e normalmente viene eseguito) attraverso il software di calibrazione fornito con la ECU. In rari casi, così come è prevista la pubblicazione di alcune variabili, il software della centralina può prevedere anche la ricezione di alcuni parametri attraverso il protocollo di comunicazione CAN. Tale caratteristica viene sfruttata per permettere al sistema di gestione sala di comandare alla centralina degli offset o dei guadagni da applicare a specifici parametri di controllo motore. Nel capitolo relativo alle automazioni di cella, questo argomento verrà approfondito, evidenziando l'importanza di tale funzionalità nei cicli di calibrazione.

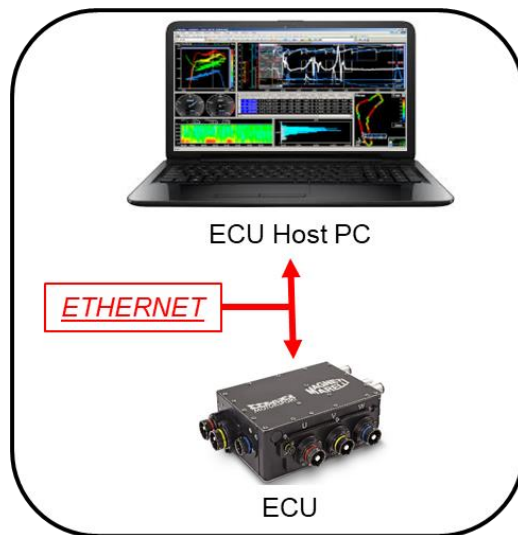


Figura 1-4 Centralina di controllo motore con relativo software di calibrazione

2. La gestione del banco

Sul mercato sono presenti molte aziende che si propongono come costruttori di banchi prova per svolgere attività di sperimentazione su oggetti di vario tipo (motori a combustione interna, motori elettrici, pompe, riduttori, motoriduttori, ecc..). Solitamente si tratta di aziende che fanno del testing il proprio core business e che quindi sviluppano internamente attrezzature e strumentazione per fare dell'attività sperimentale. Tali componenti nascono facilmente interfacciabili tra loro, dunque è naturale che queste ditte presentino sul mercato anche “turn-key” solutions, ovvero sale prove complete. Diverso è il caso di aziende che propongono soluzioni assemblate, ovvero che costruiscono banchi prova integrando diversi componenti sulla base delle esigenze del cliente, con notevoli vantaggi dal punto di vista della personalizzazione hardware e software rispetto al caso precedente. Tuttavia, il fatto di offrire un prodotto non standardizzato può comportare problemi e rallentamenti durante la messa in opera del banco e alti costi di manutenzione in seguito.

Dunque, se da una parte abbiamo un prodotto robusto e collaudato, ma rigido e poco personalizzabile, dall'altra parte ne abbiamo uno estremamente flessibile che ovviamente comporta i rischi dovuti ad una produzione non standardizzata ma cucita sulle esigenze del cliente. La vera differenza tra le due soluzioni, a prescindere da quella che è la qualità della strumentazione utilizzata per il banco, sta nel sistema di gestione della sala prove.

2.1 Requisiti

La gestione di una sala prove non è cosa banale. Come già detto in precedenza le maggiori complessità derivano dal dover interfacciare tra loro strumenti di diversa tipologia, coordinare le attuazioni con precisione implementando sia controlli specifici sia automazioni (per mantenere un elevato livello di accuratezza minimizzando i tempi delle prove) e infine assicurare la sicurezza di personale e componenti. Ciò detto, è opportuno valutare correttamente la scelta della piattaforma hardware e dell'ambiente di sviluppo software per evitare di incorrere in limiti insormontabili o che possono essere aggirati solo tramite una pesante riprogrammazione del codice.

I principali requisiti hardware e software, che deve avere un sistema di questo tipo, consistono in:

- Potenza di calcolo tale da consentire di svolgere operazioni anche complesse con frequenze che possono arrivare ad 1 KHz. Nell'ottica di svolgere test di durata considerevole, magari non presidiati, il sistema deve essere in grado di lavorare anche per tante ore in totale autonomia, senza bloccarsi o manifestare problemi dovuti, ad esempio, a saturazioni di memoria (il memory leakage è una tipica causa di malfunzionamento di software che può manifestare il suo effetto solo dopo molte ore di funzionamento);
- Le interfacce hardware e la programmazione del software devono permettere al supervisore di interagire con tutti i dispositivi della sala (possibilità di dialogare con devices di diversa tipologia sfruttando quei bus di comunicazione tipici dell'ambito automotive: CAN, seriale, ethernet);
- Essere quanto più possibile flessibile e personalizzabile lato software, e modulare ed espandibile, lato hardware.
- Consentire il riutilizzo di codice già sviluppato in linguaggi di programmazione diversi (ad esempio, integrazione di driver scritti in linguaggio di basso livello per la comunicazione con i devices, o riutilizzo di algoritmi di calcolo o modelli implementati in Matlab o Simulink);
- Avere una struttura compartimentale, in maniera tale che gli eventuali add-ons implementati dallo sviluppatore e integrati nel codice, se affetti da problemi non vadano a inficiare la corretta esecuzione del resto del software.
- Offrire allo sviluppatore degli strumenti di debug che consentano di effettuare in tempi brevi l'identificazione delle aree problematiche del codice per apportare prontamente le modifiche necessarie. Questa caratteristica può ridurre enormemente i costi di manutenzione.
- Non richiedere conoscenze di programmazione di basso livello;

- Offrire all'utente finale, l'operatore di cella, un'interfaccia chiara, leggibile e intuitiva;
- Essere facilmente aggiornabile, per seguire l'evoluzione degli strumenti e stare al passo con le release della piattaforma di sviluppo software, mantenendo la compatibilità con le versioni precedenti;
- Avere costi contenuti.

2.2 Layout hardware

Visti i precedenti requisiti relativi alla potenza di calcolo, si è scelto di impiegare una struttura Real-Time come piattaforma hardware per il sistema di gestione della sala prove. La peculiarità di un sistema di elaborazione Real-Time consiste nel fatto che esso è in grado di eseguire in modo affidabile un'applicazione con requisiti di temporizzazione molto specifici e stringenti [5]. Questa è una caratteristica fondamentale nell'ambito della sperimentazione, soprattutto quando l'oggetto delle prove è un motore a combustione interna ed è necessario produrre dati con frequenze paragonabili a quelle di combustione del propulsore. Inoltre, l'utilizzo di un sistema di questo tipo è necessario nel momento in cui è indispensabile assicurare una certa affidabilità di esecuzione del codice implementato anche per lunghi periodi di attività. Ovviamente, il componente chiave è il *Sistema Operativo Real-Time (RTOS)*. I sistemi operativi per uso generico (ad esempio il classico Windows installato sulla maggior parte dei nostri pc) sono ottimizzati per eseguire simultaneamente una varietà di applicazioni, garantendo a ciascuna di esse del tempo necessario per l'elaborazione. Ciò implica che, se sono presenti task a priorità differente, le attività ad alta priorità possono comunque essere interrotte o rallentate dall'esecuzione di altre a priorità più bassa (scansione antivirus, aggiornamento della grafica, esecuzione di attività in background), rendendo impossibile la garanzia di uno specifico tempo di risposta, e quindi il determinismo, per le applicazioni più critiche. Per quei programmi che devono essere eseguiti a una determinata velocità e senza interruzioni, come il sistema di gestione della sala prove, questo ritardo può essere catastrofico. Al contrario, i sistemi operativi Real-time sono sviluppati per eseguire un singolo programma con una temporizzazione ben precisa e, se tale applicazione comporta l'esecuzione di più task, i sistemi Real-Time offrono la possibilità di assegnare priorità alle

attività in modo che il task più critico possa sempre assumere il controllo del processore quando necessario, garantendone l'esecuzione (e quindi il determinismo). Sul mercato sono diverse le aziende d'informatica che propongono sistemi operativi Real-Time: data la conoscenza pregressa dei sistemi di acquisizione dati National Instruments, si è scelto di affidarsi ai prodotti di questa azienda, sia per la parte hardware che per la parte software. L'azienda propone sul mercato diverse tipologie di hardware per lo sviluppo di sistemi embedded dedicati al controllo e all'acquisizione di dati. Sulla base della potenza di calcolo necessaria è possibile scegliere tra processori più o meno prestazionali (diverse frequenze di clock). Dopodichè i sistemi vengono corredati delle schede di I/O specifiche per l'applicazione da sviluppare: ce ne sono di tantissime tipologie, ciascu modulo offre un condizionamento integrato del segnale specifico per un certo tipo di sensore.

Alcuni processori possono anche integrare un chip FPGA (Field Programmable Gate Array) embedded, collegato direttamente a ciascun modulo I/O e connesso al processore Real-Time attraverso un bus PCI ad alta velocità. L'architettura così definita, mostrata in Figura 2-1, garantisce alte prestazioni, affidabilità tipiche dei sistemi industriali e completo accesso alle risorse hardware di basso livello, grazie proprio al chip FPGA che può essere facilmente riprogrammato in linguaggio grafico (tramite Labview), per implementare il condizionamento software dei segnali acquisiti e avanzate funzionalità di temporizzazione, sincronizzazione, controllo ed elaborazione per l'I/O, sia analogico che digitale.

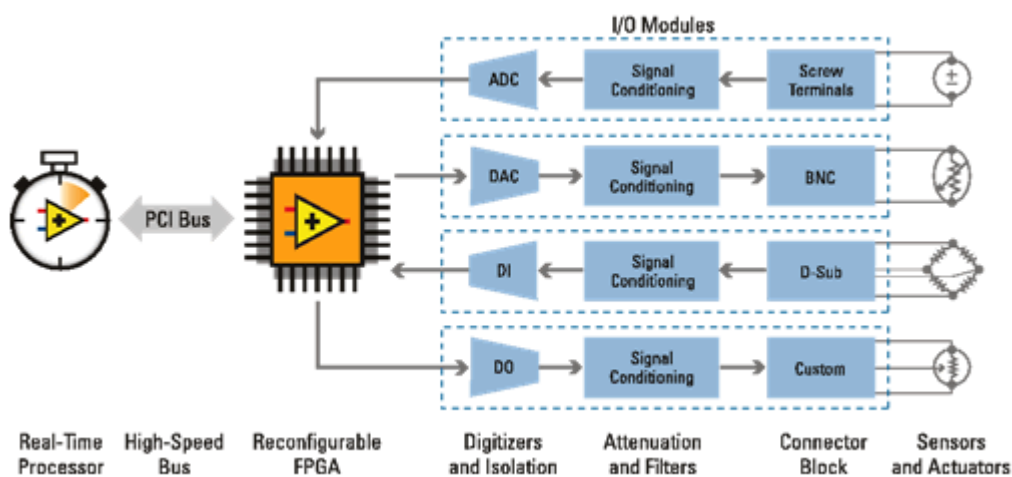


Figura 2-1: Architettura sistema real-time embedded

National Instruments fornisce sistemi embedded con l'architettura appena descritta in differenti forme per venire incontro alle diverse esigenze di compattezza e robustezza dettate dall'ambiente in cui si troverà a lavorare il sistema. Nel caso in cui l'ambiente di lavoro non sia particolarmente critico e non si sia vincolati a spazi piccoli per l'applicazione, è possibile sfruttare un comune desktop PC (che rispetti alcuni requisiti fondamentali), come processore per il sistema embedded, convertendo il sistema operativo originale in Labview RTOS. Questa soluzione offre notevoli vantaggi in termini di costo mantenendo allo stesso tempo prestazioni elevate. Il processore NI viene rimpiazzato dal desktop PC, che verrà collegato attraverso Bus MXI EXPRESS ad uno chassis, il quale integra il chip FPGA e su cui è possibile installare i vari moduli I/O.

I dati acquisiti costituiscono un flusso che dai moduli I/O arriva al processore RT attraverso il chip FPGA: le informazioni possono essere utilizzate per svolgere eventuali elaborazioni in tempo reale. C'è la possibilità di salvare i dati in file di acquisizione direttamente sul HD del target RT (in base alla tipologia di processore RT utilizzato si può disporre di un disco rigido dotato di una certa quantità di memoria utilizzabile per installare i software necessari all'esecuzione dell'applicazione o appunto per fare il data-logging). Ovviamente alcune delle informazioni acquisite devono essere disponibili a schermo per permettere all'operatore di cella di monitorare e gestire l'esecuzione della prova. Tuttavia, un grave limite dei sistemi Real-Time è che non offrono un'interfaccia utente (se il processore RT dispone di un'interfaccia video, collegandovi uno schermo si può visualizzare solo un'interfaccia a riga di comando che fornisce qualche informazione sullo stato di esecuzione dell'applicazione), per questo è necessario prevedere l'invio dei dati dal processore RT ad un computer Host, per fornire all'operatore le informazioni necessarie tramite indicatori grafici. L'interfaccia utente contiene anche i controlli (tasti e bottoni) attraverso i quali l'operatore può gestire le varie attuazioni della cella, quando non vengono già gestite automaticamente dai controlli integrati all'interno del software di sala prove. Il pc Host, oltre ad essere indispensabile per poter permettere all'operatore di gestire la sala prove, è utile anche per il data-logging. Il salvataggio dei dati sul pc Host offre due vantaggi, rispetto a quello sul processore RT:

- il primo consiste nel fatto che il target RT potrebbe non avere una disponibilità di memoria sufficiente per effettuare acquisizioni di durata considerevole.

- in secondo luogo, il fatto di avere le acquisizioni direttamente sul pc Host consente di eseguire analisi (lanciando ad esempio script di elaborazione dati appositamente creati) su quelle informazioni appena acquisite anche durante lo svolgimento della prova. Tale funzionalità risulta essere di grande utilità ad esempio per monitorare il trend di diverse acquisizioni, prendere decisioni e, sulla base di queste, variare l'esecuzione della prova mentre essa è in corso.

La Figura 2-2 mostra la piattaforma hardware utilizzata per sviluppare il sistema di gestione della sala prove.

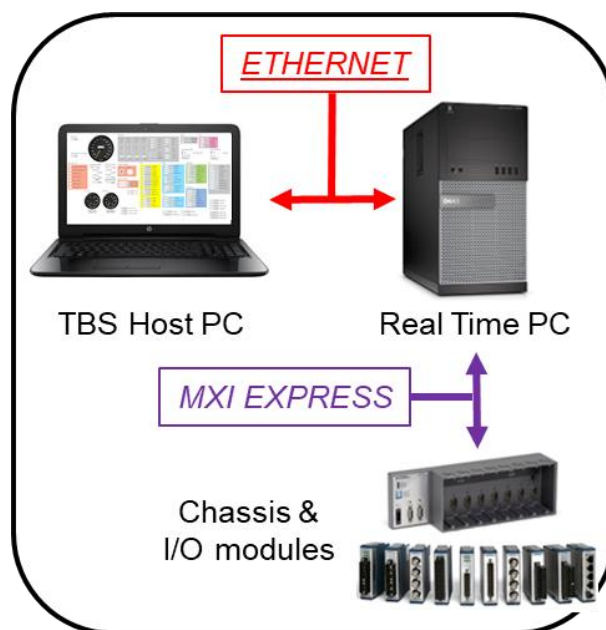


Figura 2-2: Piattaforma hardware del sistema di gestione sala prove

2.3 Ambiente di sviluppo software: NI Veristand

NI Veristand (VS) è un ambiente di sviluppo software che permette di implementare, eseguire e gestire delle applicazioni per svolgere attività di testing general purpose su piattaforme hardware Real-Time prodotte da National Instruments. Le peculiarità di NI Veristand, le quali

hanno portato a scegliere questo ambiente di sviluppo piuttosto che altri, consistono nel fatto che:

- Integra nativamente tante funzionalità di uso comune in ambito sperimentale: acquisizione (anche ad alta frequenza e con condizionamento personalizzabile) e controllo di segnali analogici e digitali, canali calcolati, linearizzazioni, allarmi e altro ancora.
- Offre all'utente un'interfaccia user-friendly, facilmente personalizzabile grazie ai tanti oggetti presenti nelle librerie native di Veristand (il pannello di comando può essere arricchito anche con oggetti sviluppati e compilati in Labview). L'interfaccia offre la possibilità del multiscreen, cioè di suddividere il pannello di controllo in più fogli.
- Laddove siano necessarie funzionalità non nativamente presenti, Veristand offre la possibilità di personalizzare il progetto creato con software compilati implementati in altri linguaggi di programmazione come Labview, Matlab o altri ambienti di modellazione di terze parti. I compilati custom possono essere utilizzati per arricchire l'interfaccia (*Custom Tools* o *Workspace Custom Objects*), aggiungere funzionalità come lo sviluppo di protocolli di comunicazione per dialogare con sensori o devices (*Custom Devices*), integrare logiche o algoritmi per effettuare controlli, simulazioni o calcoli all'interno dell'applicativo (*Models*) eseguito in Real-Time.
- Forse la caratteristica più importante: Veristand non richiede nessuna conoscenza di programmazione, a meno che non si voglia personalizzare il progetto con dei software compilati in altri ambienti di sviluppo.

Sul target hardware Real-Time viene installato il NI Veristand Engine, che gestisce le applicazioni per svolgere attività di testing. Attraverso il System Explorer, un'interfaccia Windows-based, si configurano i parametri del NI Veristand Engine e la struttura dell'applicazione che esso gestirà. Dunque viene creato un file di configurazione, detto System Definition File (SDF), che viene scaricato (deploy), via Ethernet, all'interno del target, lanciando così il NI Veristand Engine e mettendo in esecuzione l'applicazione creata.

All'atto del deploy, sul pc Host, si apre automaticamente l'interfaccia utente, anche detta Workspace, tramite cui è possibile interagire con l'applicativo in esecuzione.

Come mostra la Figura 2-3, riassumendo si può dire che NI Veristand si compone di tre parti: il *System Explorer* e il *Workspace* con cui rispettivamente si configura e ci si interfaccia con il *NI Veristand Engine*, il quale gestisce le applicazioni sul target Real-Time [6]. In realtà NI Veristand integra un ulteriore componente, utilizzato per configurare delle sequenze che possono essere eseguite dall'applicazione: lo *Stimulus Profile editor*, che sarà oggetto del sesto capitolo, relativo alle automazioni.

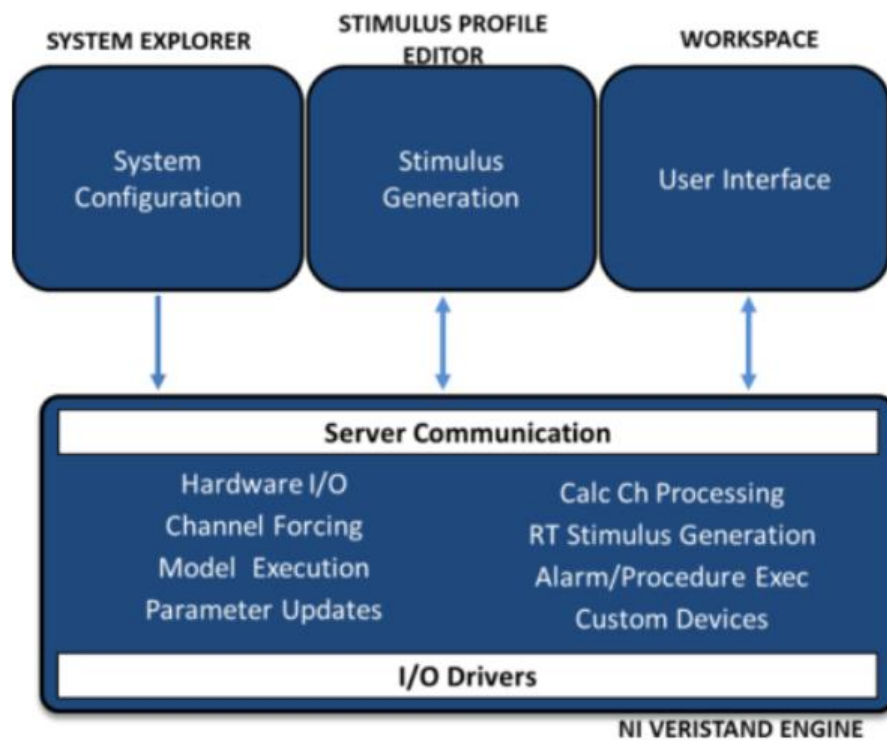


Figura 2-3: Layout di Veristand

2.3.1 NI Veristand Engine

L'Engine di NI Veristand presenta, come mostra la Figura 2-4, un'architettura costituita da diversi cicli temporizzati, la cui esecuzione temporale viene controllata con risoluzione pari al microsecondo. L'architettura multiloop consente al NI Veristand Engine di sfruttare a pieno le potenzialità di calcolo in parallelo degli attuali processori multicore [7].

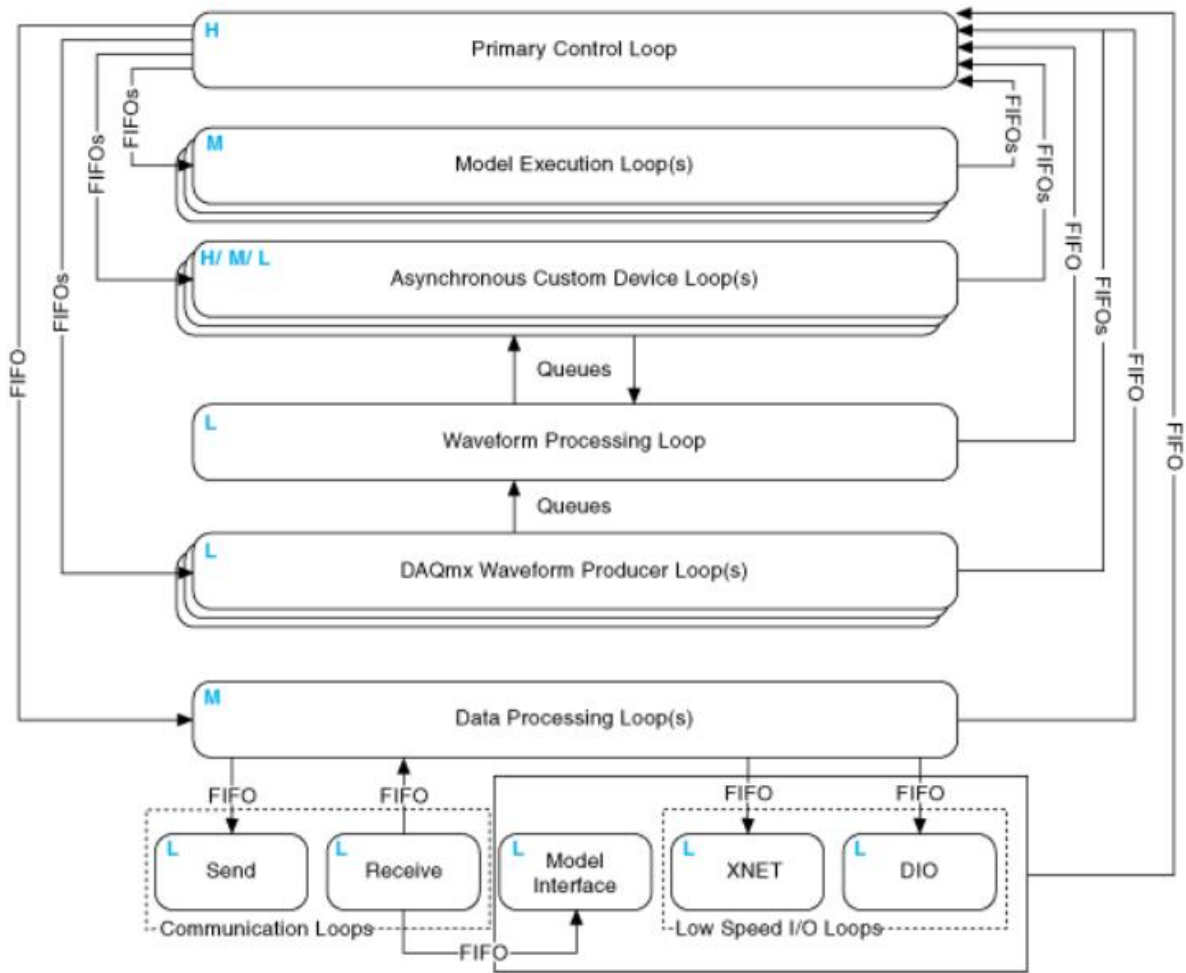


Figura 2-4: Struttura NI Veristand Engine

La comunicazione tra i cicli viene gestita tramite dei buffer di tipo FIFO. Ogni ciclo ha una priorità fissa (fatta eccezione per i Custom Devices “asynchronous”), come mostra la Tabella 2-1, mentre la frequenza di esecuzione di alcuni di essi può essere impostata in base alle esigenze dell’applicazione.

Tabella 2-1: Priorità e frequenze (di default) di esecuzione dei loop del Engine di NI Veristand

NI VeriStand Engine Loop	Priority	Default Execution Rate
<u>Primary Control Loop</u>	H	100 Hz
<u>Model Execution Loop(s)</u>	M	Decimation of the Primary Control Loop rate
<u>Asynchronous Custom Device Loop(s)</u>	H/M/L	User defined
<u>Waveform Processing Loop</u>	L	Event Driven
<u>DAQmx Waveform Producer Loops</u>	L	10 Hz; can be defined by user
<u>Data Processing Loop</u>	M	Decimation of the Primary Control Loop rate
<u>Communication Send Loop</u>	L	15 Hz
<u>Communication Receive Loop</u>	L	Event driven
<u>XNET Loop</u>	L	100 Hz
<u>DIO Loop</u>	L	Decimation of the PCL rate
<u>Model Interface Loop</u>	L	Event driven

Il ciclo principale (eseguito con la massima priorità) è il *Primary Control Loop* (PCL) che controlla l'esecuzione temporale dell'intera struttura del NI Veristand Engine, occupandosi del continuo aggiornamento del valore di ciascun canale. Ad ogni iterazione il Primary Control Loop esegue i seguenti task:

- legge e scrive ad alta velocità tutti i segnali I/O, a prescindere dal fatto che siano o meno gestiti tramite il chip FPGA, e i segnali prodotti o necessari ai Custom Devices di tipo “asynchronous”;
- applica a tutti i segnali I/O le linearizzazioni configurate nel System Definition File;
- esegue un passo di un automazione eventualmente in corso;
- invia i dati al Data Processing Loop, al fine di sincronizzare i valori di ciascun canale;
- invia i dati al Model Execution Loops;
- invia richiesta di ricezione dati a Data Processing Loops, Model Execution Loops e a Custom Device Loops;
- passa i dati tra i canali di cui si è configurato il link (mapping settings);

- Esegue sui canali richiesti gli eventuali fault configurati da interfaccia;
- Esegue i Custom Devices di tipo “inline”;

Per ogni modello che viene integrato all'interno del progetto viene creato un loop che farà parte del Model Execution Loops. Ogni ciclo riferito ad un singolo modello compie, ad ogni iterazione, le seguenti azioni:

- apprende i dati inviati dal Primary Control Loop e li assegna ai rispettivi canali di ingresso del modello sulla base di come sono stati definiti i mappings nel System Definition File;
- esegue un passo di modello;
- invia al Primary Control Loop i dati generati dal modello.

Per quanto riguarda l'esecuzione dei Custom Devices, quelli “inline” sono eseguiti ad ogni passo di Primary Control Loop, mentre gli “asynchronous” vengono gestiti in maniera completamente svincolata dal PCL (che si occupa solo di inizializzarli e di effettuare lo scambio dati con essi): ognuno è un loop a se stante con propria priorità, frequenza di esecuzione e comportamento.

Il Data processing Loop si occupa di

- eseguire le procedure e far scattare gli allarmi configurati nel System Definition File;
- elabora i canali calcolati;
- oltre ad interfacciarsi con il PCL, gestisce lo scambio dati con:
 - il Communication Loop, il quale gestisce la comunicazione TCP/IP tra pc Host e target RT;
 - il XNET Loop, che gestisce le comunicazioni di tipo CAN, LIN o FlexRay;
 - il DIO Loop che gestisce il flusso di segnali digitali I/O a bassa velocità.

Infine il Model Interface Loop si occupa dell'aggiornamento a bassa priorità dei parametri dei vari modelli nel progetto, mentre DAQmx e Waveform Loops gestiscono l'acquisizione e la generazione di una ben specifica tipologia di segnale, la forma d'onda.

2.3.2 System Explorer

Il System Explorer (SE) è l'ambiente che consente di configurare i parametri del NI Veristand Engine e di strutturare l'applicazione che si desidera eseguire sul target RT. In altre parole, la finestra del SE permette di configurare il System Definition File: si aggiungono elementi all'albero del SDF, visibile nella sezione sinistra del SE, dopodichè i parametri del singolo elemento possono essere configurati attraverso la rispettiva pagina che, una volta selezionato l'elemento specifico, viene mostrata sulla destra della finestra del SE.

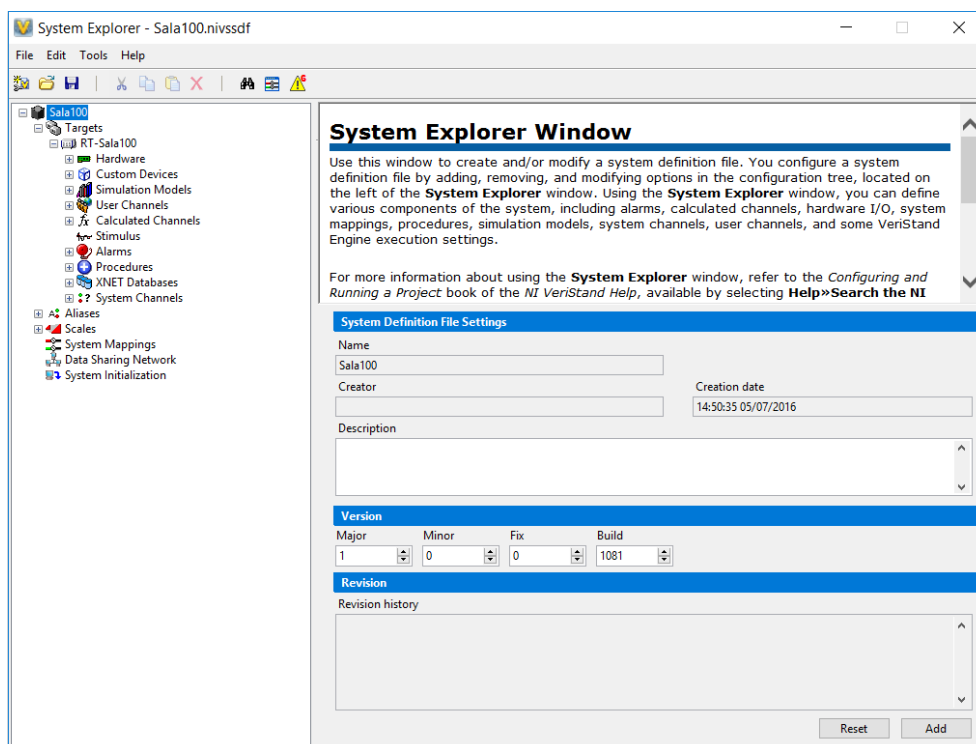


Figura 2-5: System Explorer

Più nello specifico, l'albero del System Definition File è suddiviso nelle seguenti sezioni:

- **Targets:** in questa sezione è possibile configurare tutto ciò che riguarda ciascuna applicazione gestita da un singolo target RT tra quelli che sono presenti nel progetto

(Veristand può mettere in esecuzione e gestire più processori Real-Time contemporaneamente). In particolare, per ciascun target aggiunto al progetto, l'albero del System Explorer prevede, oltre alla pagina delle impostazioni del controller RT, le seguenti sezioni:

- Hardware: in questa sezione, suddivisa in tre diverse aree, è possibile configurare tutte le comunicazioni relative ad un target.

Nella prima area si aggiungono i moduli I/O che non vengono gestiti tramite il chip FPGA, anche detti DAQ.

La seconda invece è proprio quella relativa ai segnali FPGA: per definire i moduli e i rispettivi segnali, in questa categoria è necessario indicare un file XML di configurazione che, oltre a fornire al System Explorer le informazioni necessarie per aggiornare l'albero con le schede e i segnali I/O, richiama un bitfile, compilato in Labview, che definisce il comportamento dell'hardware FPGA. Infatti, come già detto in precedenza, le interfacce I/O connesse al chip FPGA possono essere personalizzate e riprogrammate, tramite Labview, a seconda delle esigenze dell'applicazione, implementando il condizionamento più opportuno e modificando il comportamento (Personality) dell'hardware FPGA.

Infine nella terza area si configurano le comunicazioni CAN, LIN e FlexRay. Nel caso in esame, dopo aver opportunamente compilato un Database CAN, sono state configurate tre porte CAN per gestire il dialogo (sia in lettura che in scrittura) con la ECU e con il sistema di analisi combustione. Un'ulteriore porta CAN può essere utilizzata per gestire degli attuatori (come anche la macchina elettrica frenante) attraverso protocollo di comunicazione CANopen.

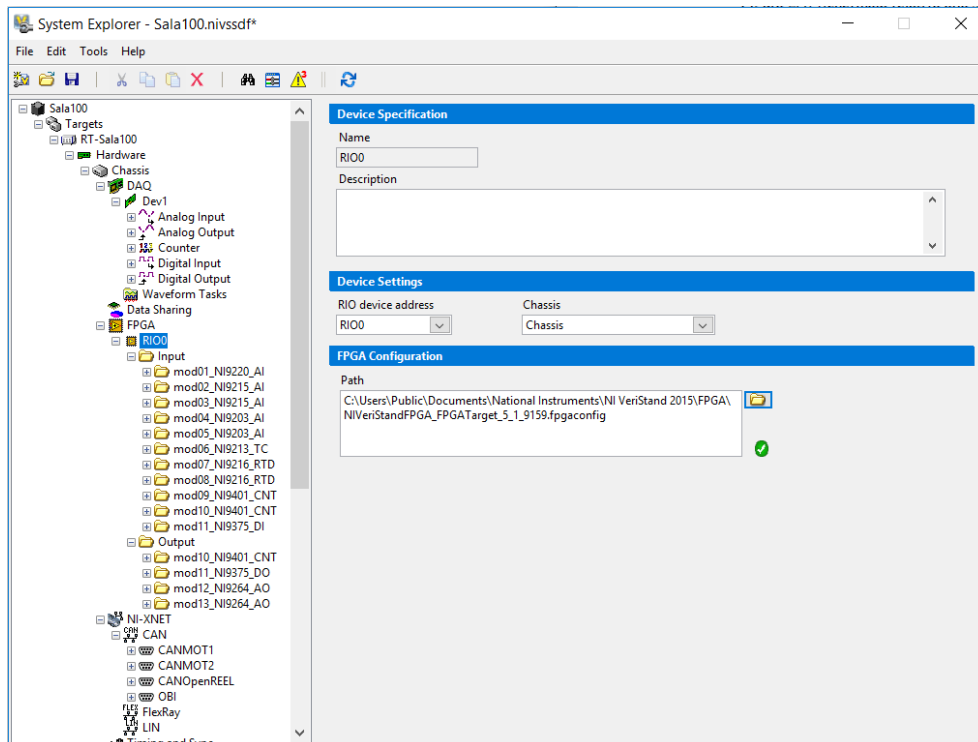


Figura 2-6: Hardware

- Custom Devices (CDs): questa sezione raccoglie i Custom Devices (compilati sviluppati in Labview per aggiungere funzionalità al progetto) di tutti i tipi (asynchronous, inline hardware interface e inline model interface), che sono stati aggiunti al progetto. Essendo ogni Custom Device un oggetto completamente personalizzabile, sia l'aspetto delle pagine dei singoli componenti che la struttura all'interno dell'albero può differire da un CD all'altro. Nella Figura 2-7 viene mostrato come esempio la Main Page di uno dei tanti componenti custom realizzati per ampliare le funzionalità di NI Veristand: il Custom Device per la comunicazione con protocollo seriale RS 485.

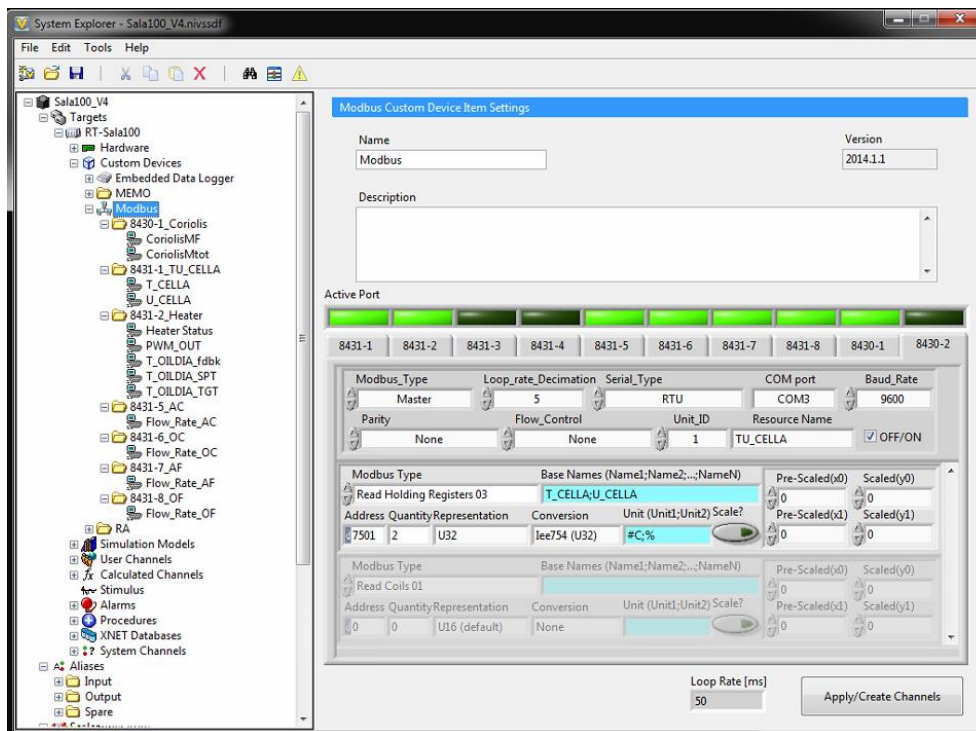


Figura 2-7: Custom Devices

- Simulation Models: i vari modelli sviluppati in altri linguaggi di programmazione e poi compilati vengono aggiunti in questa area. E' possibile indicare un file di configurazione dei parametri che verrà poi richiamato all'atto del deploy per inizializzare correttamente tutti i modelli. L'interfaccia con il modello funge da vera e propria black box che mette a disposizione i canali di ingresso e uscita nell'albero del SDF. Dopodichè, una volta che il modello compilato è stato aggiunto al progetto, le uniche opzioni che possono essere configurate sono la frequenza di esecuzione, lo stato al deploy (in esecuzione o fermo) e il processore su cui eseguirlo.

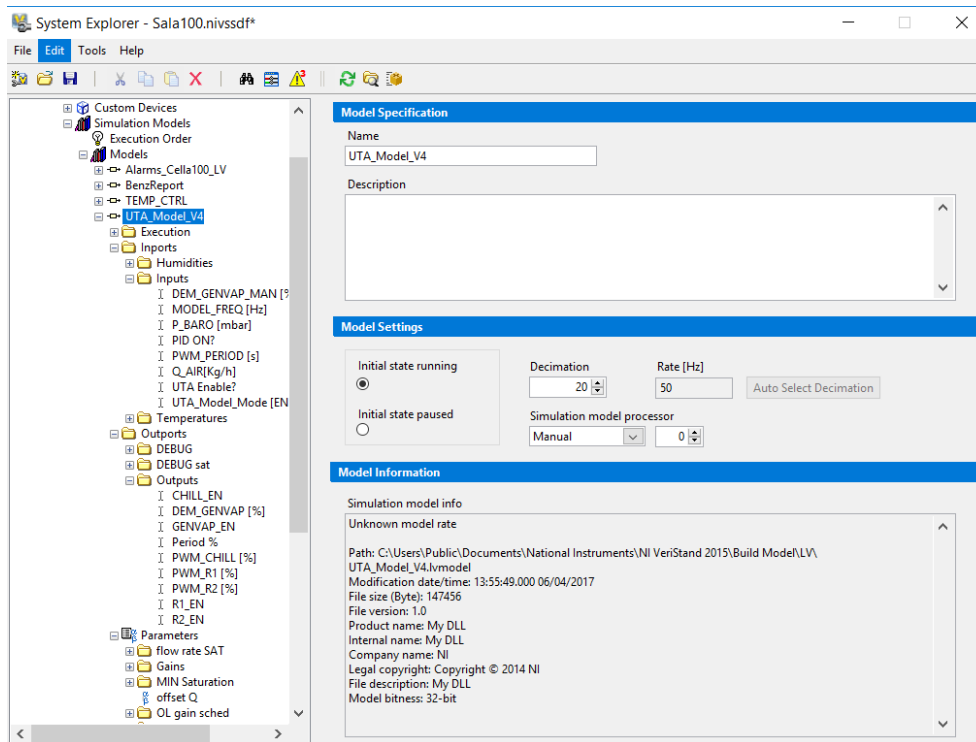


Figura 2-8: Models

- User Channels: i canali “user” vengono utilizzati per allocare dei valori costanti, come dei parametri fissi, e sono utili quando è necessario richiamare tali parametri ad esempio nelle formulazioni di canali calcolati. Un tipico user channel è il canale in cui è memorizzato il valore della cilindrata del motore (parametro utilizzato in varie formule).
- Alarms: questa sezione permette di configurare gli allarmi, specificando il canale da monitorare, le soglie da rispettare (che possono essere costanti o dinamiche), l’eventuale permanenza temporale dell’evento di superamento della soglia che deve precedere l’innesco dell’allarme, lo stato iniziale dell’allarme, le azioni da intraprendere, o meglio la procedura da eseguire, se l’allarme scatta. Inoltre, un parametro fondamentale è la priorità dell’allarme: gli allarmi possono essere divisi in gruppi, all’interno dello stesso gruppo, se più allarmi scattano in istanti ravvicinati, può essere eseguita solo una procedura, cioè quella che ha priorità più alta (la quale può eventualmente interrompere anche procedure a priorità inferiore precedentemente innescate). Al contrario,

se due allarmi appartenenti a gruppi diversi scattano nel medesimo istante, entrambe le procedure scatenate dai due allarmi verranno eseguite in parallelo, a prescindere dalla priorità assegnata.

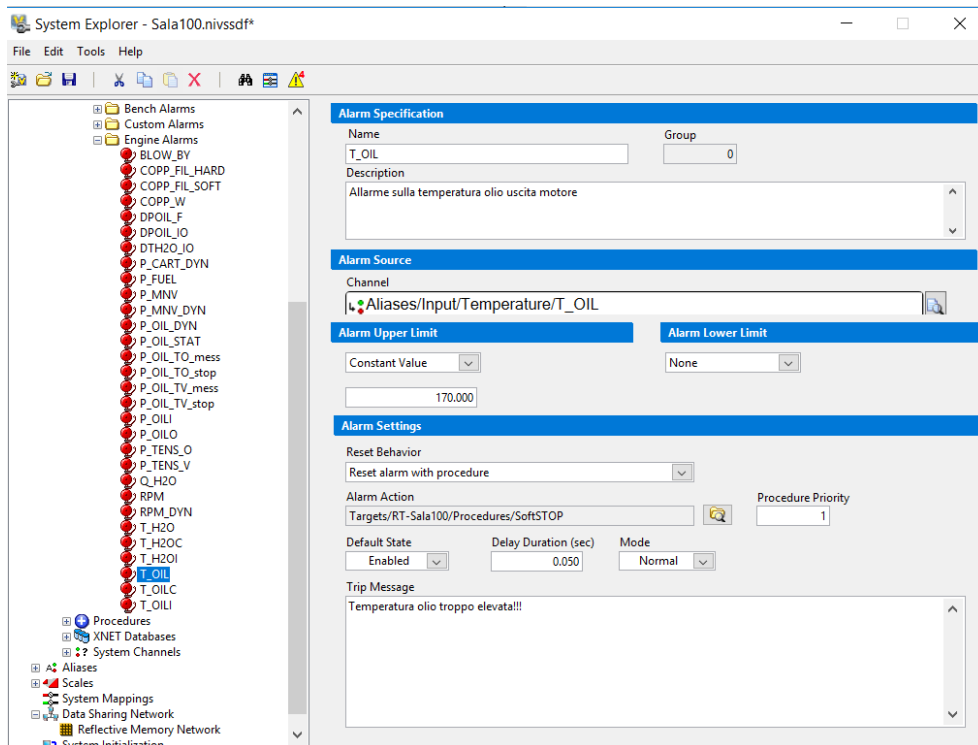


Figura 2-9: Alarms

- Procedures: le procedure sono delle sequenze di comandi, molto basilari, che possono essere richiamate quando scatta un allarme. Inoltre è possibile indicare una procedura che viene eseguita all'atto del deploy (inizio dell'esecuzione). Le azioni con cui è possibile popolare la lista di comandi permettono di uscire da eventuali sequenze automatiche in corso, agire sugli allarmi (abilitazione, disabilitazione e reset) e impostare il valore di uno o più canali contemporaneamente. Quando una procedura viene eseguita, il Data Processing Loop scorre la lista di comandi dall'alto verso il basso. E' possibile costruire dei semplici cicli con istruzioni che valutano una certa condizione e, in base al risultato, permettono alla procedura di proseguire, eseguendo il comando successivo, oppure fanno retrocedere l'esecuzione ad un'istruzione precedente. Ciò è quello che viene fatto anche nell'esempio di procedura in

Figura 2-10, in cui l'istruzione "CheckRPM" valuta il segnale DEM_DYNO_SPEED: se esso è superiore a 100 la procedura viene reindirizzata al comando precedente "Ramp", al contrario può proseguire con l'esecuzione di "Reset". Questo ciclo, in questo esempio specifico, serve ad imporre una rampa per diminuire gradualmente il regime di rotazione a seguito dello scatenarsi di un allarme che richiede l'arresto del motore (SoftSTOP).

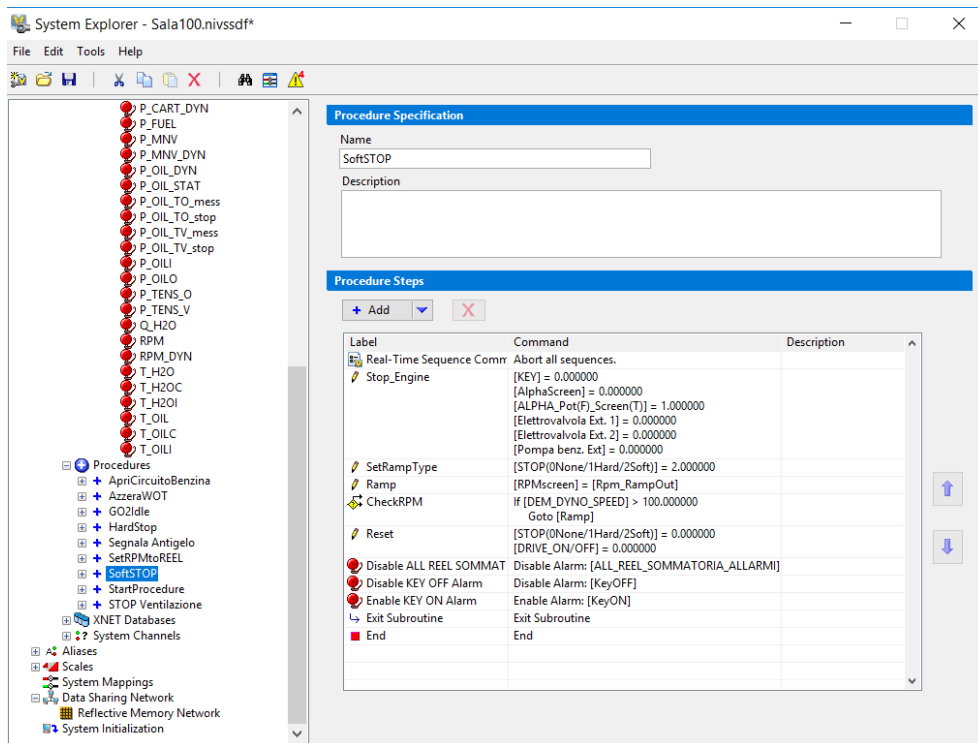


Figura 2-10: Procedures

- XNET Databases: tramite questa finestra è possibile indicare i database che si desidera utilizzare per la comunicazione via bus CAN. Attraverso un tasto presente in questa sezione si accede rapidamente al tool NI-XNET Database Editor, lo strumento fornito da National Instruments per creare Database di questo tipo.
- System Channels: è una lista di canali che Veristand mette a disposizione dell'utente per monitorare lo stato dell'esecuzione dell'applicazione sul processore RT (frequenza istantanea di esecuzione del PCL, durata dei loop

principali dell'Engine, numero di loop chiusi in ritardo e tante altre informazioni).

- Aliases: ai fini di una gestione dei canali più semplice e intuitiva, risulta essere molto comodo creare un Alias che riporta lo stesso valore immagazzinato nel canale associato ma ha un nome diverso, più leggibile e intuitivo. Tale funzionalità, apparentemente banale, risulta essere di grande comodità quando si ha a che fare con progetti molto vasti, che integrano componenti sviluppati in ottica general purpose, in cui il nome dei canali è generico e poco intuitivo (tipo mod04_NI9203_AI02), come mostra anche la Figura 2-11, mentre il nome dell'Alias può essere molto più specifico e intuitivo (ad esempio, in questo caso, Q_H2O).

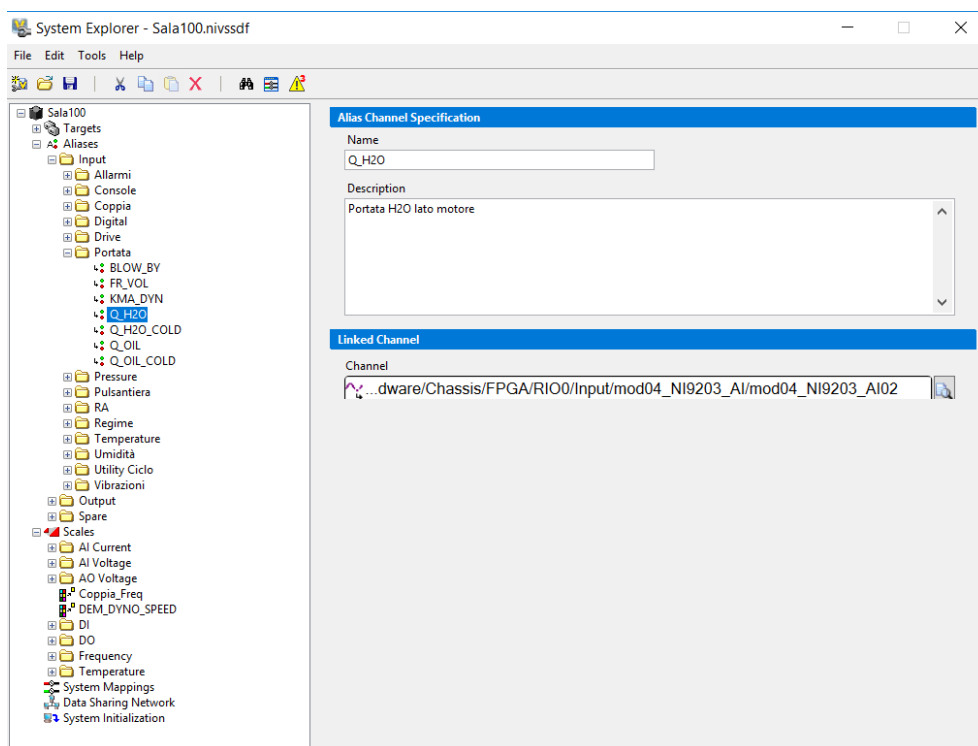


Figura 2-11: Aliases

- Scales: è la sezione in cui vengono configurate le linearizzazioni (date dalle caratteristiche dei sensori acquisiti), che poi, tramite un'apposita mappa, vengono associate ai singoli canali I/O. Una volta associata la linearizzazione, sia il canale raw

che l'eventuale Alias associato riporteranno il valore in unità ingegneristiche e non più elettriche.

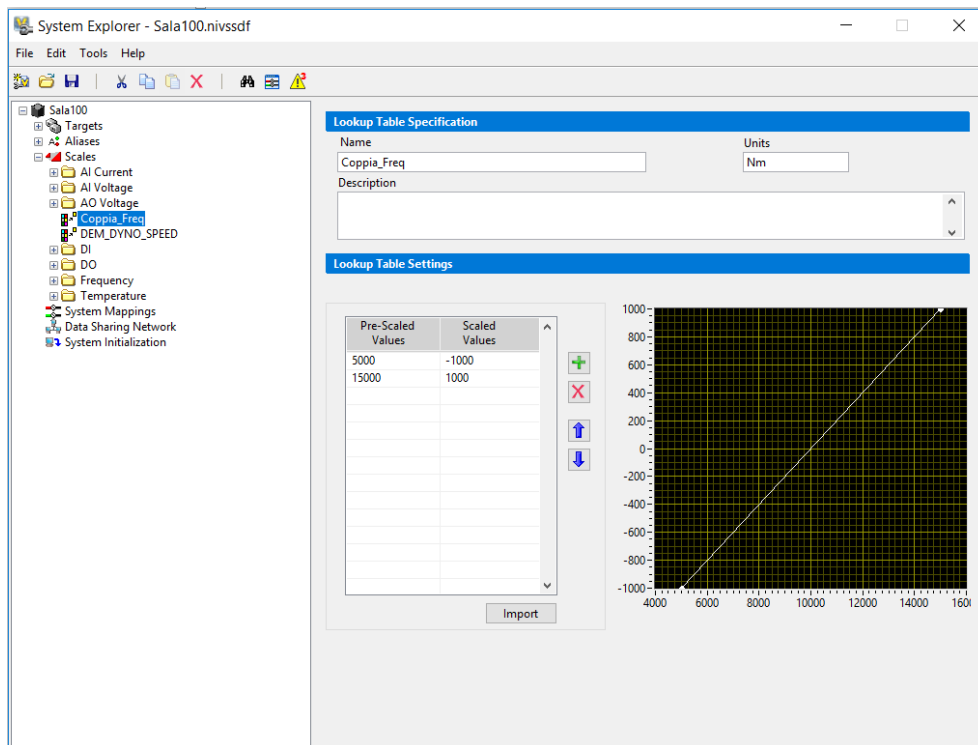


Figura 2-12: Scales

- **System Mappings:** questa sezione offre la possibilità di stabilire delle connessioni tra i canali del progetto. Infatti, i vari componenti custom che vengono integrati nel progetto sono come delle black box a cui ci si può interfacciare tramite i canali che mettono a disposizione in input e output: l'integrazione di tali componenti nell'applicazione è possibile collegando i vari ingressi e uscite delle black box con i canali I/O dell'hardware o di altre sezioni dell'albero. Una funzionalità molto utile di questa sezione consiste nella possibilità di salvare la mappa delle connessioni ed eventualmente ricaricarla in futuro.

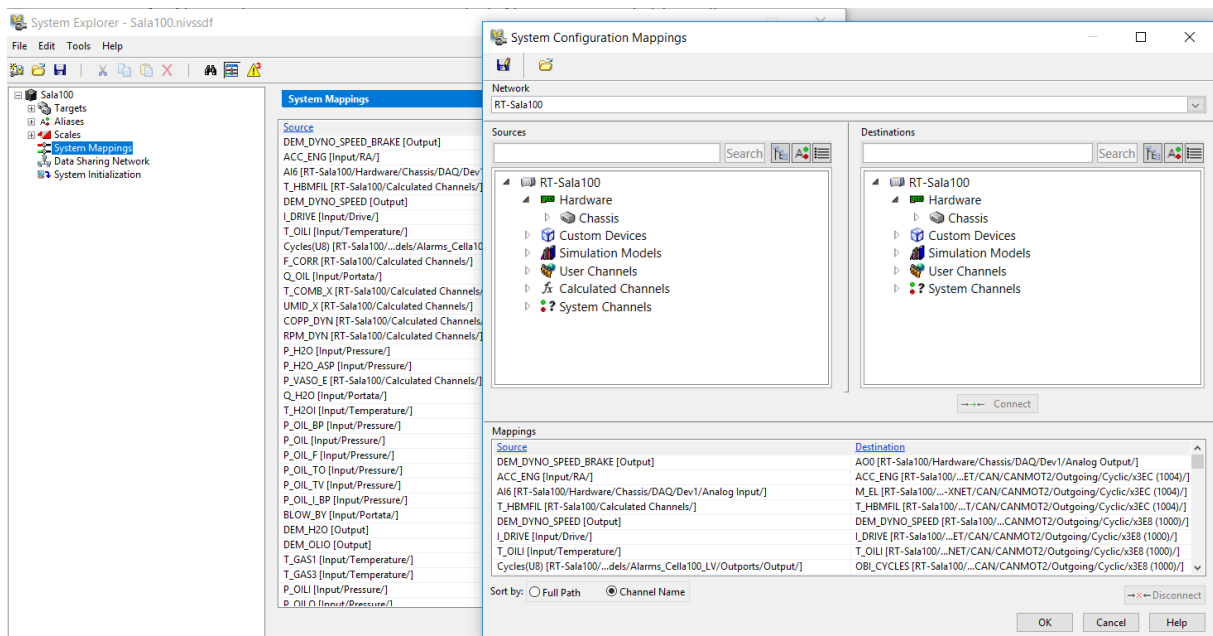


Figura 2-13: System Mappings

- **Data Sharing Network:** questa sezione permette di configurare una memoria di rete per lo scambio di dati tra i diversi target che un singolo progetto Veristand può gestire contemporaneamente. In particolare, è necessario stabilire gli indirizzi di memoria e la dimensione dinamica dei dati allocati.
- **System Initialization:** è la sezione che permette di gestire l'ordine di avvio dei target RT, nel caso in cui il progetto ne controlli più di uno contemporaneamente.

2.3.3 Workspace e UI Manager

NI Veristand, come già detto in precedenza, offre all'utente la possibilità di gestire l'esecuzione dell'applicazione scaricata sul target RT attraverso un'interfaccia utente detta *Workspace*. Recentemente, le ultime versioni di NI Veristand integrano anche una seconda interfaccia, detta *UI Manager*, che, rispetto alla precedente, ha alcune funzionalità che rendono ancora più semplice la composizione del pannello di comando (gestione di più oggetti contemporaneamente, zoom, creazione rapida di oggetti a partire dai canali configurati nel SDF). Fatta eccezione per alcune peculiarità specifiche di una o dell'altra, entrambe le interfacce offrono le seguenti funzionalità:

- possono essere personalizzate (anche mentre l'applicazione è in esecuzione) con tanti diversi indicatori e controlli presenti nelle librerie di NI Veristand (il Workspace consente di arricchire il pannello di comando anche con strumenti sviluppati e compilati in Labview, i cosiddetti **Custom Workspace Objects**);
- consentono di creare più schermi per organizzare meglio il pannello di comando (UI Manager consente anche di separare gli schermi su più monitor);
- tramite esse è possibile gestire il data-logging sul pc Host;
- Nativamente integrano alcuni tool molto utili, tra cui:
 - un pannello per la gestione degli allarmi configurati nel System Definition File;
 - un visualizzatore rapido dei dati acquisiti (.tdms) in tabelle o grafici;
 - un pannello per gestire la calibrazione dei canali;
 - un pannello che permette di visualizzare il valore istantaneo di tutti i canali del progetto sia in unità ingegneristiche che elettriche;
 - un pannello che permette di monitorare il traffico sul bus CAN;
 - un pannello per modificare i parametri degli eventuali compilati dei modelli integrati nel progetto.
 - un pannello che permette di eseguire la sovrascrittura dei canali del progetto con valori a piacere: una funzionalità molto utile per validare e fare il debug di procedure di recovery o logiche di controllo simulando condizioni particolari.
 - Un pannello che permette di registrare la sequenza di azioni eseguite dall'operatore sull'interfaccia e poi di riprodurla rispettando la cadenza temporale delle operazioni.

E' possibile sviluppare dei **Custom Tool** in Labview da inserire tra quelli sopra elencati per aggiungere funzionalità alle interfacce, come ad esempio il monitor che mostra le acquisizioni delle webcam presenti in sala prove.

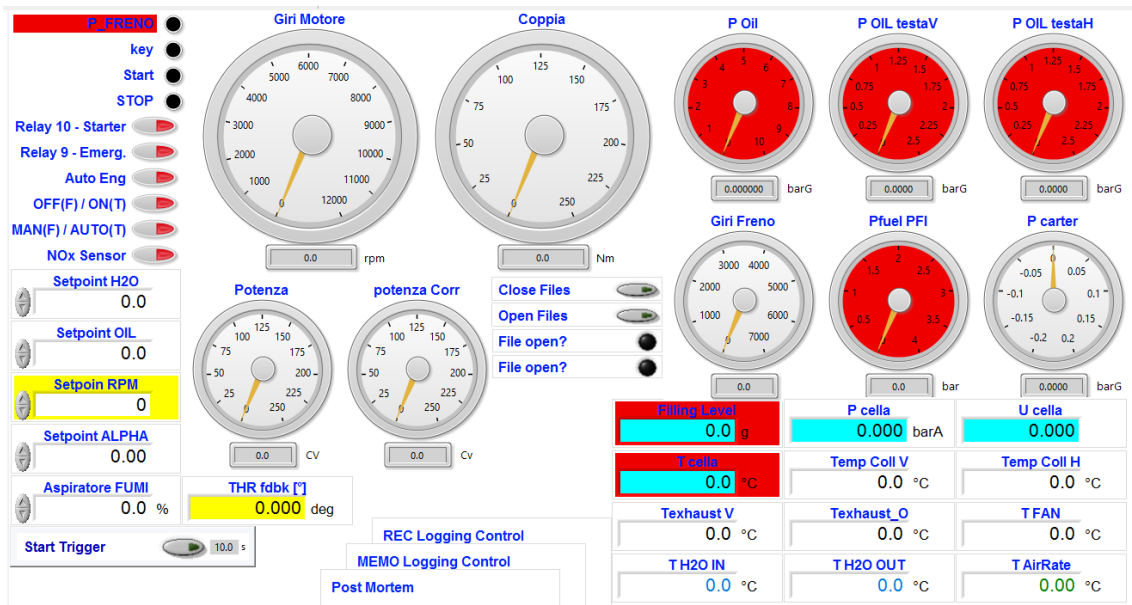


Figura 2-14: Workspace

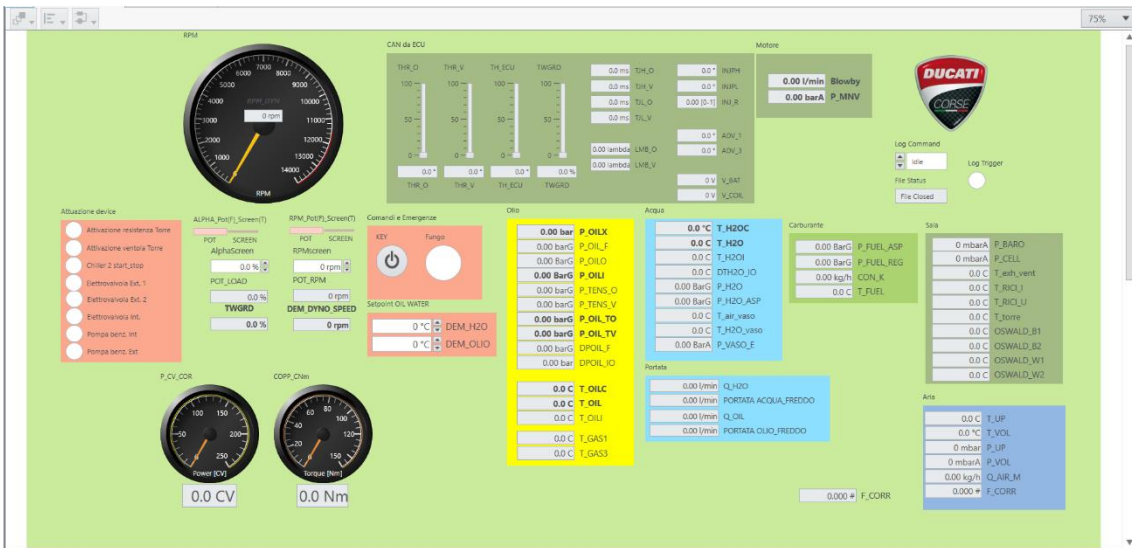


Figura 2-15: UI Manager

3. Sviluppo di oggetti custom per NI Veristand

In questo capitolo vengono descritte le modalità di implementazione di alcuni componenti custom (sviluppati in linguaggio grafico nell'ambiente di programmazione di NI Labview) compilati per essere inseriti all'interno del progetto Veristand, in modo da poter disporre di alcune funzionalità necessarie per la gestione della sala prove motore e non nativamente presenti in NI Veristand.

3.1 Fpga Personality

Nel precedente capitolo si è fatto riferimento al fatto che NI Veristand permette all'utente di personalizzare la gestione dell' I/O hardware grazie alla possibilità di poter riprogrammare, sfruttando uno specifico modulo di Labview, le interfacce hardware FPGA secondo le esigenze dell'applicazione sviluppata. Questa possibilità permette di realizzare qualsiasi genere di funzionalità in termini di condizionamento dei segnali, controllo, simulazione e anche post-processing dei dati.

Come per altre tipologie di componenti custom realizzati in Labview per NI Veristand, anche nel caso della realizzazione di un FPGA Personality si parte da un progetto template che viene configurato secondo le proprie esigenze [8]. La realizzazione delle logiche di acquisizione, controllo e condizionamento dei segnali FPGA è del tutto simile alla scrittura di un VI FPGA per generare un bitfile, che sintetizza la logica di gestione delle interfacce hardware da utilizzare in un progetto Labview. Anche in questo caso sono presenti diversi cicli eseguiti in parallelo, solitamente uno per ogni diversa tipologia di modulo, ciascuno temporizzato secondo le caratteristiche della scheda di acquisizione gestita nel loop. All'interno del ciclo è possibile implementare il condizionamento del segnale di cui si necessita. In Figura 3-1 viene mostrata la logica implementata per l'acquisizione di due moduli NI 9215 (4 canali analogici in tensione per scheda) utilizzati in sala prove per acquisire segnali derivanti da sensori di pressione olio. In questo esempio la frequenza di acquisizione è pari a 100000 campioni/secondo per canale (come dichiarato dalle specifiche del modulo) e i dati vengono condizionati attraverso un filtro Butterworth prima di esser resi disponibili all'applicazione RT.

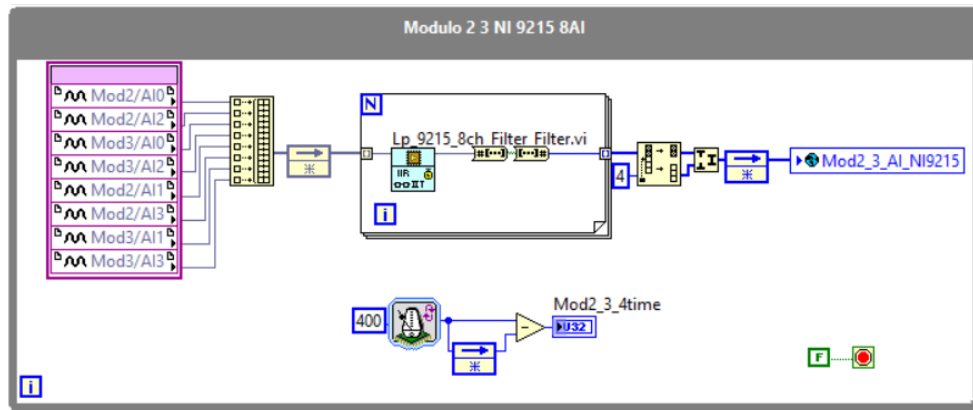


Figura 3-1: Acquisizione di due moduli NI9215

Un altro esempio di acquisizione è riportato in Figura 3-2, in cui viene mostrata l’acquisizione di segnali digitali che vengono opportunamente condizionati con un filtro integrale [28] (prima di esser giudicato vero o falso, il segnale deve effettivamente permanere in stato vero o falso per più di un certo numero di campioni) i cui parametri possono essere gestiti da NI Veristand tramite il System Explorer.

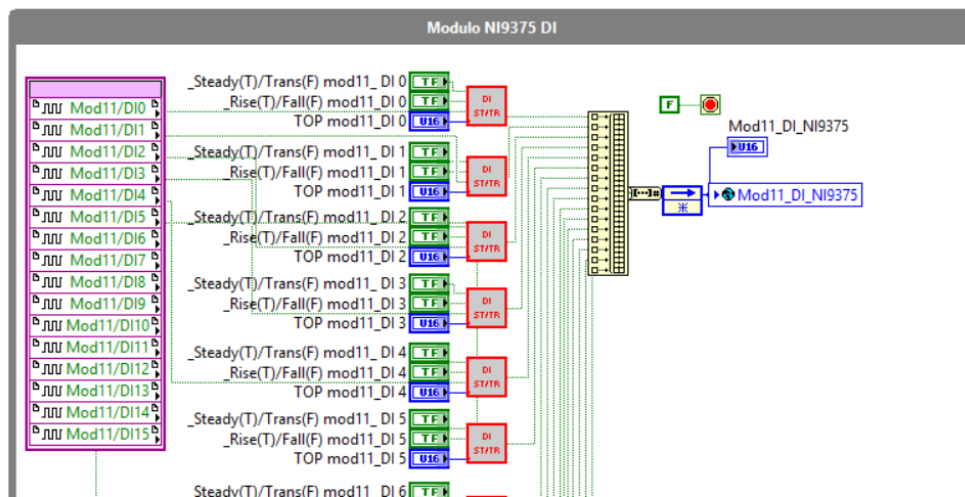


Figura 3-2: Acquisizione modulo NI 9375

Sebbene l’implementazione della gestione di hardware FPGA sia uguale a quella implementata per un normale progetto Labview, una Personality sviluppata per essere utilizzata con Veristand deve integrare anche una logica che permetta la sincronizzazione tra l’applicazione RT e il chip FPGA. Inoltre deve essere previsto un “Communication Loop”, mantenuto costantemente in sincronia con l’applicativo eseguito da NI Veristand, per gestire

il flusso di dati tra l'applicazione e il chip FPGA. La Figura 3-3 mostra il ciclo di comunicazione implementato per la personalità FPGA del progetto per la sala prove. I dati vengono trasmessi e ricevuti rigorosamente in pacchetti da 64 bit, attraverso dei buffer di memoria FIFO (first-in-first-out) detti DMA. I buffer di memoria sono configurati in modo da avere una dimensione esattamente pari al numero di pacchetti da trasferire ad ogni iterazione. Da notare il fatto che il primo pacchetto in scrittura da FPGA verso l'applicazione non può essere utilizzato poiché serve a comunicare all'applicazione se il precedente ciclo FPGA si è chiuso in ritardo o meno. Nell'esempio riportato in Figura 3-3 si nota che sono presenti ben 65 pacchetti in scrittura verso l'applicazione RT (DMA read) e 21 pacchetti in lettura dal FPGA (DMA write). In questo caso specifico l'hardware I/O della sala prove è composto da 13 moduli, di cui 9 utilizzati per acquisizione, 2 per il controllo e due moduli divisi a metà tra acquisizione e controllo di segnali digitali.

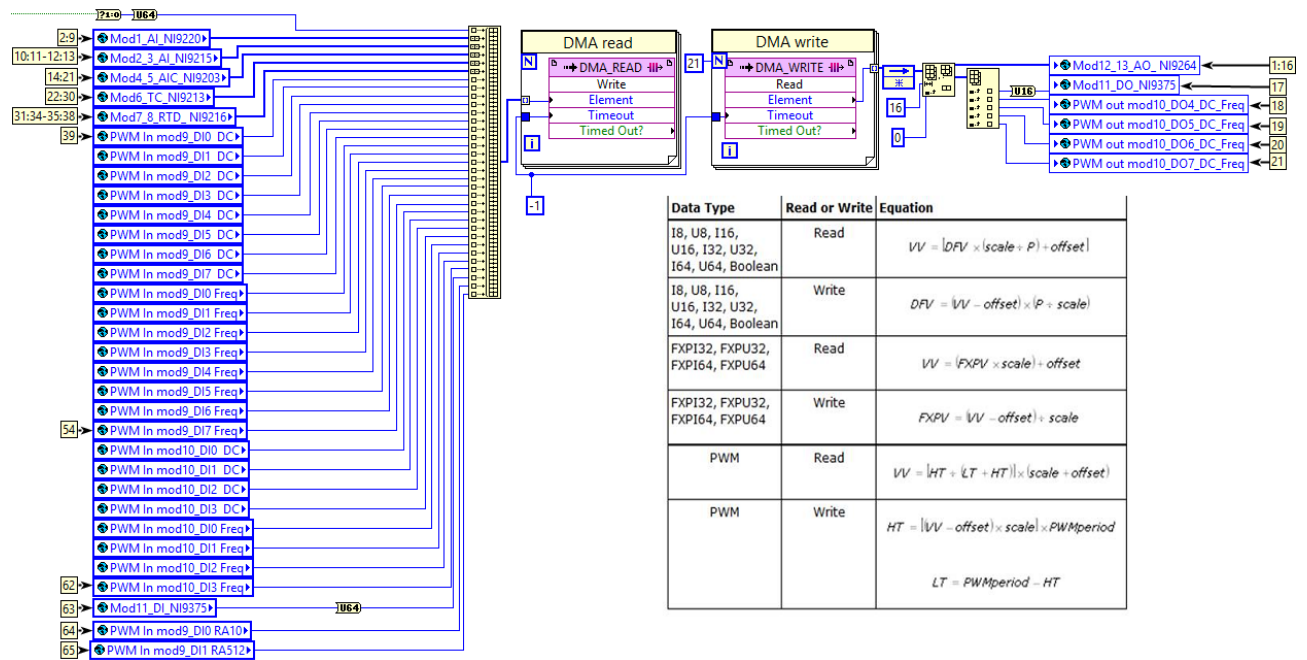


Figura 3-3: FPGA Personality: Communication Loop

Infine, per permettere alla personalità FPGA creata di interfacciarsi con l'applicazione sviluppata in NI Veristand è necessario generare un file XML (fpgaconfig) che descrive come l'eseguibile RT e la Personality devono interagire tra loro. In particolare, all'interno di questo file deve essere dichiarato il numero di pacchetti presenti nelle DMA FIFO, sia in lettura che in scrittura, inoltre devono essere specificati tutti i moduli e i relativi segnali I/O di cui deve

disporre l'applicazione gestita da NI Veristand. Per ogni segnale che viene dichiarato è possibile inserire anche i parametri del relativo condizionamento: queste informazioni verranno riportate nella pagina del segnale sul System Explorer, da cui sarà possibile impostare i parametri del condizionamento del canale. Anche per quanto riguarda il file XML esistono diversi esempi che possono essere riadattati alla propria necessità. La Figura 3-4 mostra un esempio di configurazione di un segnale digitale (NI 9375 DI00) sul file XML fpgaconfig: oltre a definire il segnale vengono dichiarati i parametri del condizionamento che viene eseguito sul canale (filtro integrale a cui si è fatto riferimento nell'esempio di Figura 3-2) in modo da poterli impostare sul SE.

```

<!--modulol1 NI9375-->
<Packet> <!--Packet 63-->
  <Boolean>
    <Name>mod11_NI9375_DI00</Name>
    <Description>Digital line DI00 on the NI-9375 in slot 11.</Description>
    <Category>Input\mod11_NI9375_DI</Category>
    <Symbol>DI</Symbol>
    <Parameters>
      <Boolean>
        <Name>Steady(1) - Transient(0)</Name>
        <ControlName>_Steady(T)/Trans(F) mod11_DI 0</ControlName>
        <InitialValue>1</InitialValue>
        <Scale>1</Scale>
      </Boolean>
      <Boolean>
        <Name>Normal/Rising (1) - Negate/Falling (0)</Name>
        <ControlName>_Rise(T)/Fall(F) mod11_DI 0</ControlName>
        <InitialValue>1</InitialValue>
        <Scale>1</Scale>
      </Boolean>
      <U16>
        <Name>Digital Filter [samples]</Name>
        <ControlName>TOP mod11_DI 0</ControlName>
        <InitialValue>10</InitialValue>
        <Scale>65535</Scale>
      </U16>
    </Parameters>
  </Boolean>
</Boolean>

```

Figura 3-4: Esempio di configurazione di un segnale in fpgaconfig

3.2 Models

I modelli sono l'add-on di più facile realizzazione per NI Veristand. Come ricordato in precedenza, possono essere implementati in diversi linguaggi di programmazione, non solo Labview ma anche Simulink (MathWorks), GT-POWER (Gamma Technologies), AMESim

(LMS), C/C++ e tanti altri. Il processo di creazione in Labview è ovviamente il più semplice di tutti: gli altri linguaggi di programmazione richiedono l'installazione (a volte un po' macchinosa) di compilatori specifici. In sostanza si crea banalmente un VI (Virtual Instrument) in Labview, che implementa la logica desiderata (calcoli statistici, modelli di simulazione motore o componenti vari, ecc.). Dopodichè, se si dispone di una licenza Labview che contiene il Model Interface Toolkit, è possibile compilare il modello generando un file ".lvmodel" che può essere aggiunto nell'albero del System Definition File. Come mostra la Figura 2-8, il modello mette a disposizione in due differenti cartelle i segnali in ingresso (inports) e quelli in uscita (outports) che permettono di interagire con il compilato. Ovviamente nel modello creato è possibile definire dei parametri, che si differenziano dagli input poiché tendenzialmente costanti rispetto a questi ultimi. I parametri di ogni compilato compaiono nell'albero del SDF in una cartella diversa da quella degli input: ciò è possibile grazie al fatto che su Labview gli input vengono contrassegnati come ingressi "required" mentre i parametri come ingressi "recommended". I parametri dei modelli, come già ricordato nel capitolo precedente, possono essere inizializzati all'atto del deploy da un file di testo indicato nel SDF. Mentre l'applicazione è in esecuzione è possibile cambiare il valore dei parametri con uno dei tool dell'interfaccia (Model Parameter Manager), che permette anche di salvare e ricaricare un determinato file di configurazione.

Data la semplicità di implementazione, i modelli sono stati ampiamente utilizzati, durante lo sviluppo del software di gestione della sala prove, per aggiungere funzionalità al progetto Veristand, tra cui:

- controlli PID;
- gestione dinamica delle soglie di allarme;
- logiche di gestione di attuatori presenti in sala;
- contatori per il chilometraggio e il conteggio dei litri di combustibile consumati.

Questa tipologia di oggetto custom è stata utilizzata anche per implementare le logiche di controllo dei sistemi di condizionamento della sala prove che verranno descritte nel prossimo capitolo.

In Figura 3-5 viene riportato un esempio di modello implementato in NI Labview per essere integrato in NI Veristand: controllo PID. La struttura tipica di un modello prevede un'architettura single cycle loop al cui interno è presente l'algoritmo, mentre all'esterno del ciclo si trovano gli ingressi del modello, siano essi input o parametri, e le uscite.

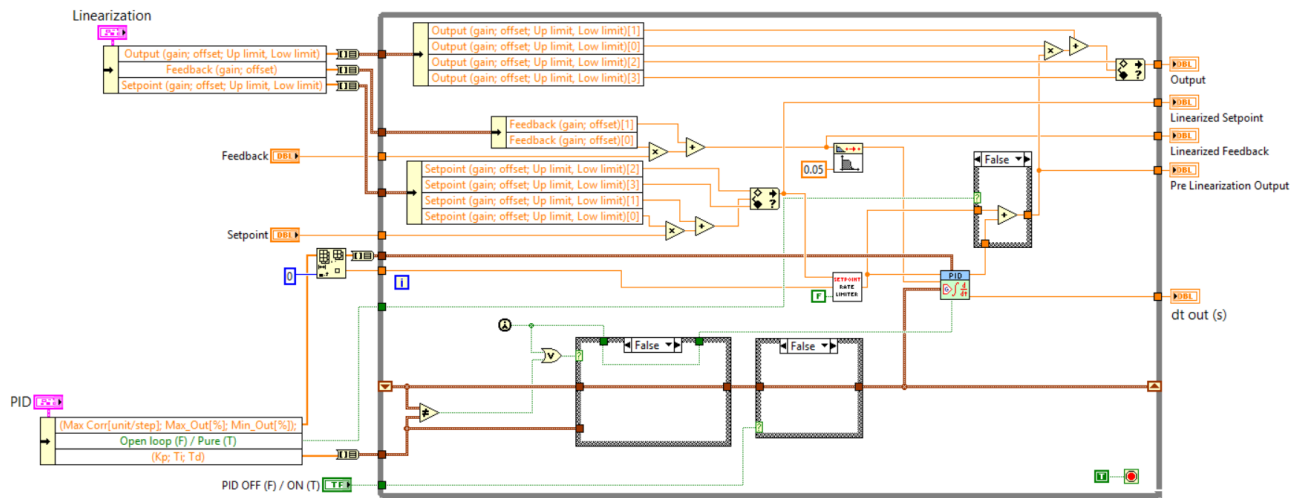


Figura 3-5: Esempio di modello implementato in NI Labview: controllo PID

La Figura 3-6 mostra un esempio di modello implementato in Simulink: gestione dinamica delle soglie di allarme. In questo caso, per rendere gli ingressi e le uscite del modello visibili nell'albero del System Definition File, è necessario utilizzare dei blocchetti specifici (in figura sono quelli di colore giallo) che permettono di interfacciarsi con Veristand: tali componenti fanno parte di una libreria che deve essere aggiunta nella raccolta di Simulink. Per quanto riguarda i parametri, per poterne disporre su Veristand è necessario dichiararli su Simulink come "inline parameters" nelle impostazioni del modello, che vanno opportunamente configurate (in particolare il solutore deve essere impostato a step fissi e non in stato continuo, con "stop time" a infinito) per permettere all'applicativo RT di gestire correttamente il compilato.

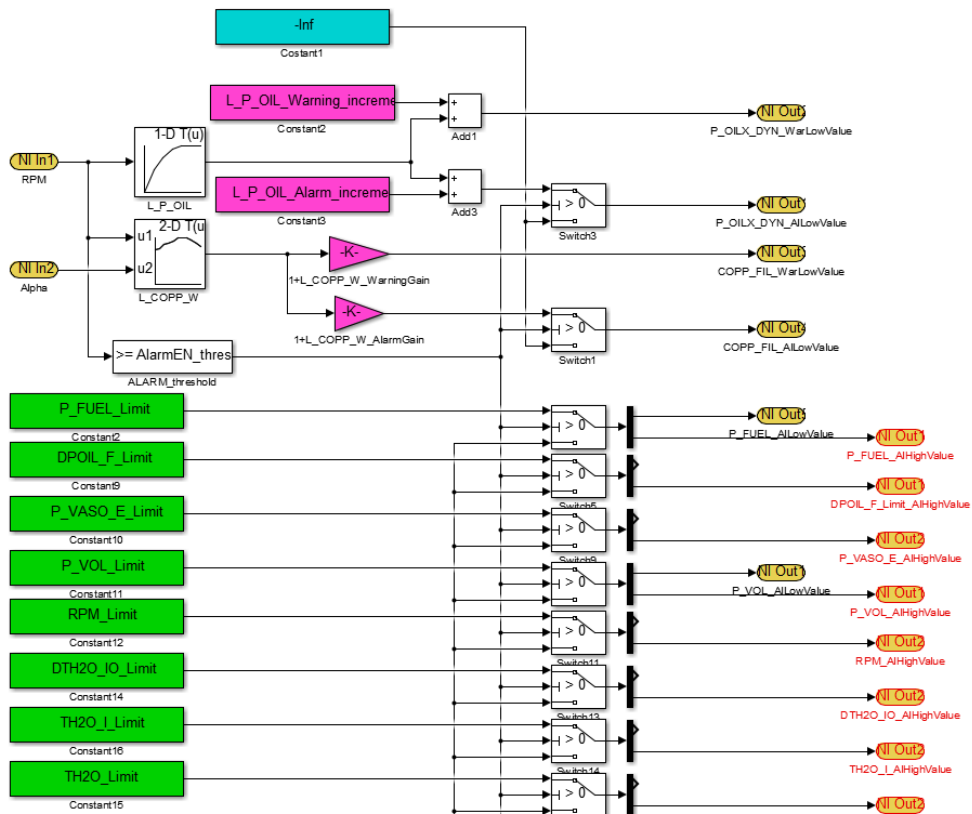


Figura 3-6: Esempio di modello implementato in Simulink: gestione dinamica delle soglie di allarme

3.3 Custom Devices

I Custom Devices (CDs) si distinguono dai modelli perché permettono di implementare funzionalità personalizzate per Veristand con una maggior libertà di sviluppo in termini di struttura del codice ed esecuzione in Real-Time, offrendo inoltre la possibilità di interfacciarsi con Veristand non solo tramite i canali messi a disposizione dal compilato, ma anche attraverso delle pagine nel System Explorer. Infatti, mentre il modello è un compilato che viene eseguito solo ed esclusivamente dall'applicazione RT, il Custom Device si divide in due parti (o meglio due librerie *.llb*): l'*Engine* consiste nel codice che viene gestito dall'applicativo in tempo reale, mentre il *Configuration* è la parte che si occupa dell'interfaccia con il System Explorer e che consiste in una serie di pagine accessibili dal SE per configurare i parametri del Custom Device.

Anche la realizzazione di un Custom Device, come quella di un FPGA Personality, richiede la realizzazione di un progetto in Labview. Solitamente lo sviluppo ha inizio da dei progetti Labview template, che vengono creati da un apposito tool fornito da National, in cui è necessario configurare alcuni parametri: destinazione e nome del progetto, modalità di esecuzione, numero di pagine extra da includere nel progetto.

Il progetto creato, come viene mostrato in Figura 3-7, oltre a contenere la libreria *Custom Device API.lvlib*, contiene:

- VI relativi alla libreria *Configuration*:
 - Initialization VI. Questa parte di codice viene eseguita quando il Custom Device viene aggiunto nell'albero del System Explorer: solitamente viene utilizzato per costruire parte della struttura del CD nell'albero, definendo le cartelle e i segnali che fin da subito devono essere disponibili nel progetto.
 - GUID Lookup VI. In questo VI sono definite le references delle pagine (sia la Main che le Extra) e delle variabili globali eventualmente utilizzate all'interno del Custom Device.
 - Main Page VI. Questa parte di codice è quella che contiene la programmazione e l'aspetto (cioè il pannello frontale) della pagina principale del Custom Device, cioè l'oggetto top-level nell'albero.
 - Extra Pages. Tramite questi VI è possibile dare una personalità (cioè fornire un aspetto e una funzionalità attraverso la scrittura di un codice) a degli oggetti che non sono la pagina principale.
- VI relativo alla libreria *Engine*, cioè il RT Driver VI, che è la parte di codice che viene eseguita dall'applicazione in Real-Time. Questa parte di codice può assumere un'architettura differente in base alla tipologia di Custom Device che si desidera utilizzare, in particolare esistono cinque diversi tipi di CD ed ognuno interagisce in maniera differente con l'Engine di NI Veristand. Durante lo sviluppo del software per la gestione della sala prove sono state sfruttate solo due tipologie di Custom Device:

- *Inline Model Interface*: questa tipologia di CD si presta molto nel caso sia necessario svolgere dei calcoli in tempo reale e il livello di interazione fornito da un modello non sia sufficiente (ad esempio nel caso in cui si desideri avere un numero di canali su cui svolgere i calcoli che può essere variabile in base alle operazioni che vengono svolte sul System Explorer). La struttura del RT Driver, in questo caso specifico, prevede una macchina a stati con quattro casi (Initialize, Start, Execute Model, Close) e il VI viene eseguito durante un'iterazione del Primary Control Loop di VS, dunque è possibile acquisire, elaborare e generare dati in tempo reale, senza ritardi. Questa tipologia di Custom Device è stata utilizzata per implementare calcoli statistici come medie temporali, medie mobili, deviazioni standard di alcuni segnali, ecc..
- *Asynchronous*: è il tipo di Custom Device che permette la maggiore libertà di sviluppo per il RT Driver VI. Inizialmente sono previsti solo due loop, il primo per gestire la ricezione di comandi (come ad esempio lo Shutdown) dal Veristand Engine e il secondo per gestire lo scambio di dati con l'applicazione Real-Time. A parte questi due cicli il resto del VI può essere strutturato a piacimento ed essendo questo CD una parte di codice che viene eseguita in maniera asincrona rispetto al Primary Control Loop, la sua esecuzione non può influenzare gli altri componenti gestiti dal Veristand Engine. Ciò fa sì che questo tipo di Custom Device si presti allo sviluppo di algoritmi per la gestione di hardware attraverso protocolli non deterministici (che potrebbero far accumulare dei ritardi al sistema se fossero eseguiti inline con il PCL), per svolgere processi in background o monitorare lo stato del sistema (operazioni a bassa priorità). Infatti, durante lo sviluppo del software per la sala prove, si è fatto uso di questa architettura per sviluppare i driver per il colloquio con alcuni devices presenti in sala prove (bilancia, riscaldatore olio, stazioni barometriche..)

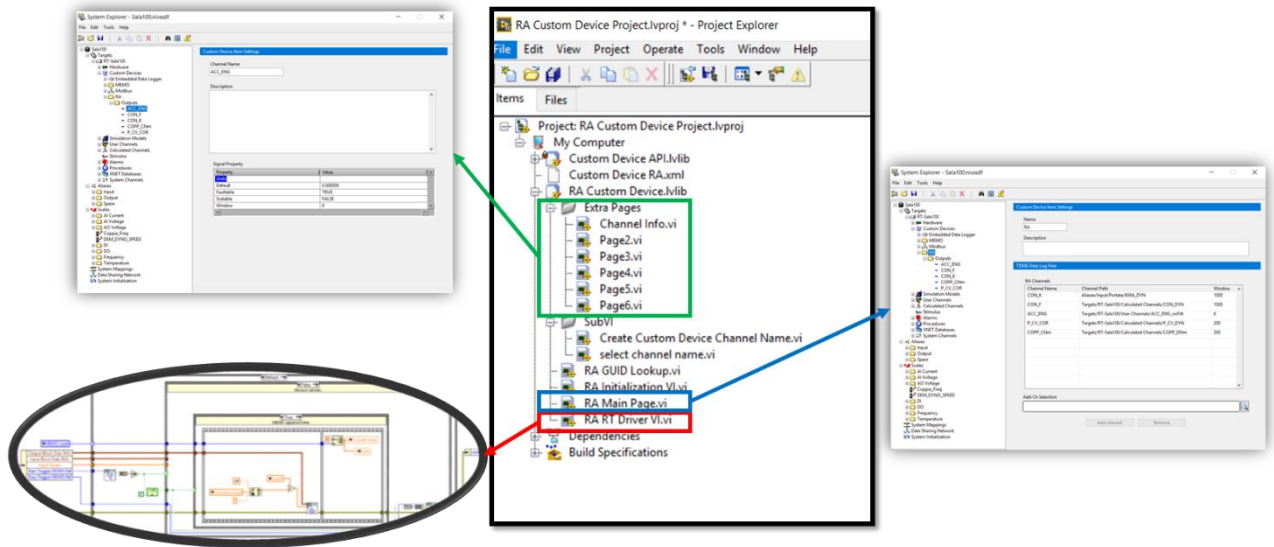


Figura 3-7: Struttura di un Custom Device

4. Il controllo dei sistemi di condizionamento del banco

Come già riportato nel primo capitolo, una sala prove motore solitamente dispone di vari sistemi preposti al controllo di alcuni fattori, che possono influenzare il comportamento del motore, più in generale, il risultato delle prove in termini di prestazioni, emissioni e affidabilità dei componenti [1, 2, 3].

Solitamente i singoli componenti di ogni sistema vengono controllati in maniera indipendente l'uno dall'altro, attraverso dei PLC (Programmable Logic Controllers). Tuttavia, un approccio di gestione centralizzato può consentire una maggior coordinazione tra le varie attuazioni, evitando che le azioni intraprese dai singoli componenti siano in conflitto tra loro. Un esempio di un controllo di questo tipo, che è stato applicato in ambito automotive con eccellenti risultati, è la gestione in coppia nei sistemi di controllo motore.

Durante l'attività di dottorato è stato sviluppato un approccio centralizzato e model-based per il controllo dei sistemi di condizionamento di temperatura olio e acqua motore e temperatura e umidità dell'aria in aspirazione. Questa metodologia di controllo è stata sviluppata con l'obiettivo di assicurare la miglior accuratezza e ripetibilità in termini di raggiungimento e mantenimento del target da parte delle variabili controllate, gestendo con coordinazione le attuazioni dei singoli componenti di ciascun sistema.

In letteratura [9, 10, 11, 12] sono tanti i lavori che mostrano come la temperatura del refrigerante motore possa essere correttamente gestita sfruttando strategie di controllo MPC (Model Predictive Control) nella gestione termica del propulsore: il problema è simile. L'approccio sviluppato è leggermente diverso dal tipico metodo MPC, infatti, invece di identificare i modelli dinamici che descrivono il processo in questione [13], vengono utilizzate equazioni di stato per descrivere le interazioni fisiche tra i vari componenti del sistema allo scopo di definire i contributi in catena aperta dell'algoritmo di controllo, mentre la dinamica dei fenomeni da gestire viene compensata tramite il contributo in catena chiusa fornito da classici controller PI. Per quanto riguarda la gestione della temperatura del lubrificante motore, è più difficile trovare degli elaborati relativi a questo argomento [14, 15] in letteratura: in questo caso si è deciso di utilizzare il medesimo approccio di controllo anche

per la gestione della temperatura olio motore e anche delle condizioni dell'aria comburente. Più nello specifico, il contributo in catena aperta viene stabilito sulla base del bilancio termico applicato al singolo componente considerato, mentre il contributo fornito dalla retroazione si basa semplicemente sulla differenza tra le condizioni del fluido desiderate e quelle effettivamente misurate.

Una volta sviluppato, il modello di controllo è stato integrato nel sistema di gestione della sala prove: ciò ha permesso di validare l'approccio implementato direttamente sul campo di utilizzo, alimentando gli input del modello con i sensori del banco prove, svolgendo prove sperimentali con il motore in condizioni sia stazionarie sia transitorie. La sala prove in questione viene utilizzata per svolgere prove di durata, calibrazione e sviluppo su motori motociclistici ad alte prestazioni.

Nei prossimi paragrafi di questo capitolo vengono descritte le strutture dei sistemi di condizionamento controllati, viene presentato l'algoritmo di controllo e infine vengono riportati i risultati dei test condotti per la validazione della metodologia di controllo.

4.1 Plant dei sistemi di condizionamento

4.1.1 Layout unità di trattamento dell'aria comburente

Le condizioni dell'aria aspirata dal motore possono essere controllate attraverso un'unità di trattamento aria (o HVAC: Heating and Ventilating Air Conditioning): una struttura a pannelli che racchiude al proprio interno diversi componenti per il trattamento dell'aria. Il dimensionamento e il design di tale sistema, ovvero la scelta dei componenti, viene opportunamente studiato per rispondere alle esigenze specifiche dell'applicazione: si deve tenere in conto la portata di aria che deve essere elaborata dal sistema, i livelli di temperatura e umidità che si vogliono raggiungere e la pressione a cui si vuole rendere disponibile l'aria all'uscita del trattamento. Solitamente l'intero sistema viene collocato o all'esterno, magari sul tetto dello stabile, oppure all'interno di un locale tecnico.

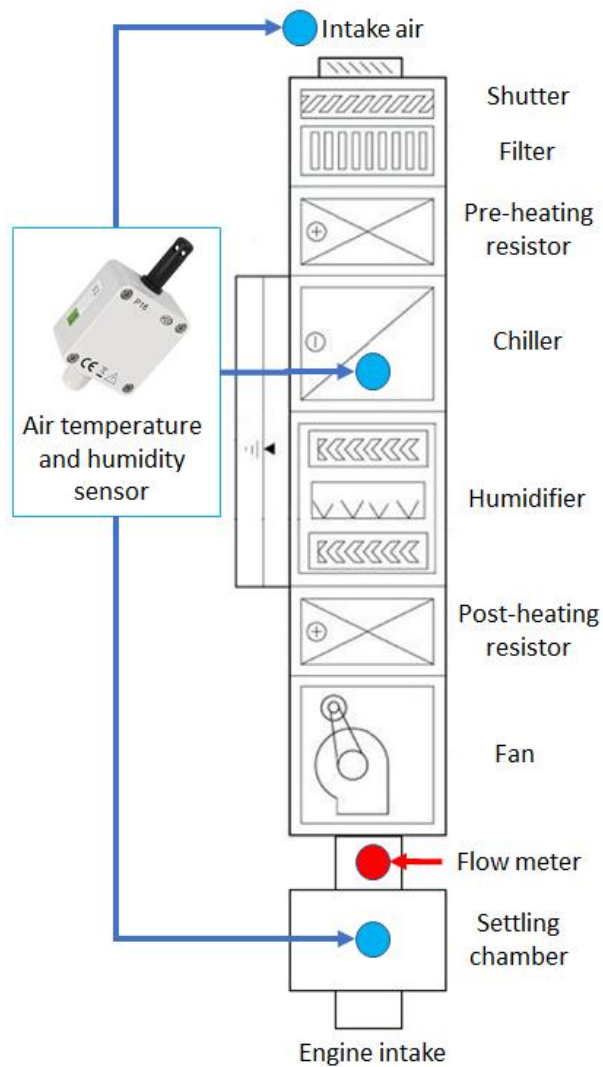


Figura 4-1: Layout unità di trattamento aria

La Figura 4-1 mostra lo schema dell'unità di trattamento aria della sala prove in cui è stato implementato il controllo descritto in questo capitolo. All'ingresso del sistema di trattamento è presente una saracinesca che permette di regolare manualmente l'afflusso di aria entrante. A valle un filtro trattiene le impurità trasportate dall'aeriforme evitando un rapido imbrattamento dei componenti successivi. Segue dunque una resistenza di pre-riscaldamento che permette di effettuare un primo incremento di temperatura dell'aria. Un termostato spegne la resistenza quando una soglia impostabile di temperatura viene superata. A valle è posizionata la batteria di raffreddamento, ovvero uno scambiatore di calore alimentato da una sorgente fredda, nel nostro caso un chiller. In questa sezione l'aria cede calore al liquido

refrigerante che scorre nel lato opposto dello scambiatore. Se la temperatura dell'aria trattata scende al di sotto del punto di rugiada, il vapore acqueo trasportato dall'aeriforme si condensa, riducendo l'umidità assoluta dell'aria. Il produttore del chiller fornisce la Tabella 4-1, che permette di stimare, sulla base della temperatura dell'aria, della del liquido refrigerante e della sua composizione (percentuale di glicole), la capacità di raffreddamento. L'acqua condensata viene solitamente raccolta in un serbatoio ed evacuata all'esterno del locale tecnico. La batteria di raffreddamento è seguita da un nebulizzatore, collegato ad un generatore di vapore, che inietta in seno all'aria del vapore acqueo allo scopo di aumentarne l'umidità. Infine, una resistenza di post-riscaldamento permette di effettuare un ulteriore incremento della temperatura dell'aria prima che essa venga spinta all'esterno dell'unità di trattamento da un ventilatore centrifugo. La velocità di quest'ultimo può esser eventualmente regolata mediante un inverter per gestire la pressione del volume di calma, tuttavia il controllo della pressione dell'aria non è argomento di questo elaborato. L'aria trattata, percorrendo un tubo in PVC, arriva dunque ad un misuratore di portata, seguito da un volume di calma collegato direttamente con l'aspirazione del motore attraverso dei tubi flessibili sigillati.

Tabella 4-1: Capacità di raffreddamento Chiller

Cooling Capacity [kW]	Tair									
	14 [°F] -10 [°C]	32 [°F] 0 [°C]	50 [°F] 10 [°C]	59 [°F] 15 [°C]	68 [°F] 20 [°C]	77 [°F] 25 [°C]	86 [°F] 30 [°C]	95 [°F] 35 [°C]	104 [°F] 40 [°C]	113 [°F] 45 [°C]
14 [°F] / -10 [°C] Glycol 30%	9,2	9,2	9,2	9,2	9,2	8,8	8,2	7,7	7,2	
23 [°F] / -5 [°C] Glycol 30%	11,4	11,4	11,4	11,4	11,3	10,7	10,0	9,4	8,8	
32 [°F] / 0 [°C] Glycol 20%	14,0	14,0	14,0	14,0	13,7	12,9	12,2	11,5	10,7	
41 [°F] / 5 [°C] Glycol 15%	17,0	17,0	17,0	17,0	16,2	15,4	14,5	13,7	12,8	11,8
44,6 [°F] / 7 [°C] Glycol 0%	18,5	18,5	18,5	18,4	17,5	16,6	15,7	14,8	13,9	12,9
50 [°F] / 10 [°C] Glycol 0%	20,4	20,4	20,4	20,1	19,2	18,2	17,3	16,3	15,3	14,2
59 [°F] / 15 [°C] Glycol 0%	23,9	23,9	23,9	23,1	22,1	21,0	20,0	18,9	17,7	16,5
68 [°F] / 20 [°C] Glycol 0%	27,8	27,8	27,4	26,3	25,1	24,0	22,8	21,5	20,0	18,5
77 [°F] / 25 [°C] Glycol 0%	29,0	29,0	28,3	26,9	25,6	24,2	22,9	21,5	20,0	18,5

L'algoritmo che controlla i vari componenti del sistema di condizionamento dell'aria viene alimentato con le quantità fisiche acquisite dal sistema di gestione del banco prove. In particolare, le informazioni necessarie riguardano le condizioni dell'aria, in termini di umidità e temperatura, che vengono monitorate all'ingresso del motore ma anche in vari punti

all'interno dell'unità di trattamento. Come mostra la Figura 4-1, vengono acquisiti tre sensori temperatura e umidità aria: il primo all'ingresso dell'unità di trattamento, il secondo è collocato a valle della batteria di raffreddamento, il terzo si trova all'interno del volume di calma. Ovviamente anche la portata massica e la pressione dell'aria sono dati utili al controllo: una stazione barometrica è posizionata all'interno del locale tecnico, mentre, come precedentemente detto il misuratore di portata è installato a valle dell'unità di trattamento aria (prima del volume di calma).

Per quanto riguarda gli output del modello, necessari al controllo del sistema di trattamento aria:

- l'umidificatore richiede un segnale analogico proporzionale al flusso di vapore e un comando di abilitazione digitale;
- resistenze e chiller sono controllati mediante segnali PWM (pulse width modulation), gestendo la percentuale di periodo di controllo in cui i sistemi sono attivi o disattivati. Sono disponibili altre opzioni di controllo (gestione diretta della corrente dei resistori mediante tiristori e controllo del chiller tramite l'impostazione da remoto della temperatura target del refrigerante) che possono determinare una maggiore dinamica di controllo, ma la struttura del controller non cambia.

Tabella 4-2: Caratteristiche tecniche componenti unità trattamento aria

Potenza resistenza di pre-riscaldamento	12 kW
Potenza di raffreddamento massima chiller	14.8 kW
Produzione di vapore massima generatore di vapore	26.4 kg/h
Potenza resistenza di post-riscaldamento	18 kW
Portata massima ventilatore centrifugo	1000 m ³ /h

4.1.2 Layout sistema di condizionamento dell'acqua motore

Il sistema di condizionamento del liquido refrigerante motore consiste in due circuiti idraulici: nel primo circola l'acqua fornita da una torre evaporativa, mentre nel secondo l'acqua motore, una miscela di acqua demineralizzata e glicole etilenico. Come mostrato in alto nella Figura 50

4-2, una pompa volumetrica spinge il fluido, prelevato dalla torre evaporativa, verso uno scambiatore di calore a piastre opportunamente dimensionato: all'interno di questo componente l'acqua riceve calore dal liquido refrigerante del motore, che scorre in controcorrente nello scambiatore. In seguito, l'acqua torna alla torre evaporativa, mentre il refrigerante, raffreddato, ritorna al motore.

Per quanto riguarda questo sistema di condizionamento, le informazioni necessarie al controllo per gestire i vari componenti sono temperature e portate di entrambi i fluidi. Per ciascun liquido viene acquisita la temperatura sia a monte che a valle dello scambiatore di calore, con delle termoresistenze (Pt100). Un sensore ad ultrasuoni (con trasduttori "clamp on" direttamente attaccati sull'esterno della tubazione) viene utilizzato per monitorare la portata di acqua di torre all'ingresso dello scambiatore di calore, mentre per quanto riguarda la misura di portata del liquido motore viene utilizzato un sensore di tipologia Vortex Shedding.

In questo caso, il modello di controllo può gestire lo scambio termico tra i due fluidi agendo solamente, con un segnale analogico, sulla velocità della pompa volumetrica, che, in quanto tale, elabora una portata che è direttamente legata alla velocità di rotazione della macchina stessa. Solitamente il sistema di condizionamento dei fluidi motore prevede una pompa centrifuga accoppiata ad una valvola di controllo della portata, tuttavia, lo schema con pompa volumetrica comporta il vantaggio di avere una relazione stabile tra il comando (la velocità) e l'effetto (lo scambio termico) indipendentemente da quelle che sono le cadute di pressione nel circuito. Inoltre le dinamiche di regolazione sono più veloci e senza isteresi.

4.1.3 Layout sistema di condizionamento dell'olio motore

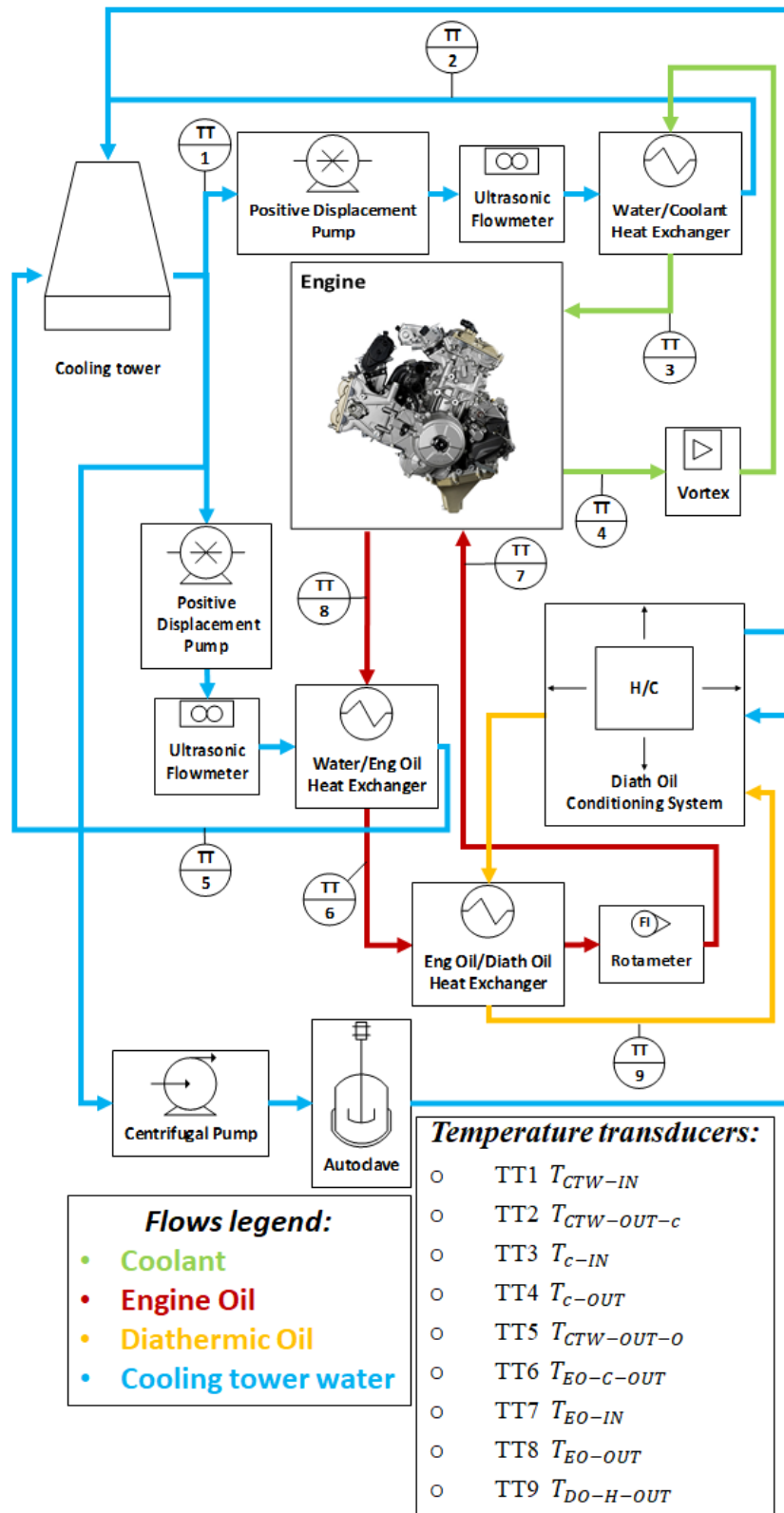


Figura 4-2: Layout sistema di condizionamento acqua e olio motore

Tabella 4-3: Caratteristiche tecniche componenti sistemi di condizionamento acqua e olio motore

SISTEMA CONDIZIONAMENTO REFRIGERANTE MOTORE	
Potenza di raffreddamento torre evaporativa	250 kW
Massima portata pompa acqua di torre	60 l/min
Capacità nominale scambiatore di calore (acqua torre/refrigerante motore)	36 kW
SISTEMA CONDIZIONAMENTO LUBRIFICANTE MOTORE	
Capacità nominale scambiatore di calore (acqua torre/lubrificante motore)	16 kW
Massima portata pompa acqua di torre	60 l/min
Capacità nominale scambiatore di calore (olio diatermico/lubrificante motore)	18 kW
Capacità nominale scambiatore di calore (acqua torre/olio diatermico)	17 kW
Potenza riscaldatore olio	18 kW
Temperatura massima olio diatermico	180 °C

Mentre il sistema di condizionamento dell'acqua motore prevede l'utilizzo di due circuiti accoppiati da uno scambiatore di calore, la temperatura del lubrificante viene gestita con un sistema più complesso, costituito da tre circuiti: questa differenza è dovuta all'esigenza di poter riscaldare l'olio motore oltre all'ovvia possibilità di poterlo raffreddare. In maniera del tutto simile a quanto visto per il sistema di condizionamento dell'acqua, anche in questo caso, un flusso di acqua della torre evaporativa scorre verso uno scambiatore di calore grazie ad una pompa volumetrica dedicata, assorbe calore per poi ritornare a raffreddarsi in torre. Nello scambiatore di calore scorre, in controcorrente, l'olio motore che, una volta ceduto calore all'acqua di torre, prima di tornare nel propulsore, fluisce verso un altro scambiatore dove avviene un secondo scambio termico, questa volta con dell'olio diatermico. La temperatura dell'olio diatermico viene gestita da un sistema di condizionamento dedicato, costituito da un riscaldatore (che sfrutta delle resistenze) e da uno scambiatore di calore interno per il raffreddamento. Quest'ultimo viene alimentato con acqua della torre evaporativa attraverso una pompa centrifuga, inoltre un'autoclave consente di mantenere una pressione pari a 2.5 bar all'interno del circuito, mentre la portata viene regolata tramite una valvola dedicata. Il vantaggio di questo schema sta appunto nella possibilità di poter anche riscaldare l'olio motore, una soluzione che permette di velocizzare molto la fase di riscaldamento del propulsore, diminuendo notevolmente lo spreco di combustibile durante questa operazione. Inoltre, il riscaldatore di olio diatermico permette di eseguire test simulando delle condizioni che non sarebbero raggiungibili senza di esso, come ad esempio prove a basso regime e carico con olio motore ad alte temperature. Infine, un ulteriore vantaggio sta nel fatto che il sistema

di condizionamento dell'olio diatermico è, come già descritto, dotato di un sistema proprio di raffreddamento. Ciò permette di raffreddare l'olio motore anche con il diatermico, che può dunque cooperare con l'acqua di torre al raffreddamento del motore, ad esempio prima dello spegnimento motore una volta terminate le prove, evitando ancora una volta inutile spreco di carburante (in questa fase, infatti, la differenza di temperatura tra acqua di torre e olio motore è limitata, quindi la potenza termica scambiata tra i due fluidi è a sua volta ridotta. A poco serve aumentare la portata dell'acqua di raffreddamento, perché in queste condizioni lo scambio termico è limitato dalla capacità termica oraria sul lato del lubrificante). Si potrebbe pensare di evitare l'installazione del sistema di raffreddamento ad acqua di torre, tuttavia il solo sistema ad olio diatermico non sarebbe efficiente nel raffreddamento dell'olio motore a causa del doppio scambio di calore (da acqua di torre ad olio diatermico e poi da olio diatermico ad olio motore) necessario per il raffreddamento. Dunque, lo scambiatore acqua di torre /olio motore viene mantenuto, assicurando un rapido abbattimento della temperatura dell'olio quando necessario.

La temperatura del lubrificante viene gestita dal modello regolando la portata di acqua inviata allo scambiatore (acqua di torre/olio motore) controllando la velocità della pompa volumetrica con un segnale analogico. Lo scambio termico con l'olio diatermico viene regolato grazie alla possibilità di gestire il setpoint di temperatura del fluido all'uscita del riscaldatore via protocollo seriale (RS 485). Il setpoint viene saturato in modo da scongiurare il pericolo che la superficie delle resistenze possa raggiungere una temperatura tale da far insorgere nell'olio diatermico a diretto contatto fenomeni di "cracking". Per lo stesso motivo si è escluso a priori di utilizzare le resistenze a diretto contatto con l'olio motore, preferendo l'olio diatermico, noto per tollerare temperature maggiori rispetto ai normali lubrificanti.

Per quanto riguarda i segnali necessari all'algoritmo di controllo, come mostra la Figura 4-2, la temperatura dell'olio motore viene misurata in tre punti: prima, dopo il motore e tra i due scambiatori presenti nel circuito. Come già visto per il circuito di condizionamento del refrigerante, la temperatura dell'acqua di torre viene acquisita prima e dopo lo scambiatore in cui fluisce. La temperatura dell'olio diatermico viene monitorata tramite termoresistenza all'ingresso del relativo sistema di condizionamento, che invece misura direttamente la temperatura del diatermico all'uscita dal riscaldatore e comunica l'informazione in seriale

(RS485). Un misuratore di portata volumetrico viene utilizzato per l'olio motore mentre il flusso di olio diatermico viene considerato circa costante poiché la pompa che mette in circolo il fluido non può essere regolata. La portata di acqua di torre viene monitorata con sensori analoghi a quelli presentati per il medesimo scopo nel circuito di condizionamento dell'acqua motore.

4.2 Algoritmi di controllo dei sistemi di condizionamento

I sistemi di condizionamento vengono solitamente controllati attraverso dei controllori di tipo PID, implementati direttamente nei PLC che gestiscono i singoli componenti. Al contrario, il controllo implementato beneficia di un contributo in catena aperta model-based, per gestire i componenti di ogni sistema con un approccio fisico, accoppiato ad un contributo in retroazione, prodotto da uno classico controllore PI. Il contributo in catena chiusa è utile per correggere le incertezze e i possibili errori contenuti nei parametri del calcolo in catena aperta [16]. Inoltre, la retroazione è necessaria dal momento che il contributo in catena aperta viene calcolato a partire da valutazioni di equilibrio statico, che non tengono conto del comportamento dinamico dei sistemi controllati. Una possibile relazione tra l'errore riscontrato e l'azione di controllo attuata per correggerlo potrebbe essere scritta come:

$$u = k_p \cdot e + k_I \cdot \int_0^t e dt + k_D \cdot \frac{de}{dt} \quad 4-1$$

Grazie al contributo fornito dal controllo in catena aperta, non è necessario un controllo in retroazione dalle dinamiche elevate, vista anche la lentezza intrinseca dei fenomeni da controllare. Ciò consente di evitare l'utilizzo del contributo derivativo (semplificando così il compito di calibrazione), solitamente impiegato per contrastare gli effetti di costanti proporzionali e integrali elevate, necessarie quando si vuole ottenere un closed-loop dalle dinamiche elevate [17].

4.2.1 *Controllo condizioni aria comburente*

Il sistema di condizionamento dell'aria comburente deve controllare fenomeni che, tra quelli controllati nell'ambito di questo lavoro, presenta la dinamica più lenta. Nello specifico, avendo a disposizione l'hardware descritto in precedenza è possibile ottenere una variazione di umidità relativa del 10 % in circa 60 secondi e una variazione di temperatura di un grado in circa 10 secondi. I parametri del controllo in catena chiusa sono stati impostati tenendo conto di tale lentezza durante i transitori, che ha reso necessario anche l'utilizzo di una saturazione della derivata delle richieste. Infatti, una variazione a gradino della temperatura o dell'umidità target dell'aria comburente porterebbe ad una rapida saturazione del contributo integrale del PI, a causa del grande errore venutosi a creare istantaneamente.

Il calcolo del contributo in catena aperta si basa sulla stima di calore e vapore d'acqua che il sistema di condizionamento deve introdurre o sottrarre dall'aria trattata, in modo da soddisfare le richieste di temperature e umidità. Per quanto riguarda la gestione dei singoli componenti:

- La resistenza di pre-riscaldamento viene utilizzata solo nel caso in cui sia richiesto un aumento di temperatura dell'aria, e al contempo non si debba diminuire l'umidità assoluta (richiesta di umidità pari al valore misurato oppure necessità di aumentare il contenuto di acqua nell'aria). L'umidità infatti viene diminuita abbassando la temperatura al di sotto del punto di rugiada per far condensare il vapore acqueo in seno all'aria: se per aumentare la temperatura in caso di deumidificazione si utilizzasse la resistenza di pre-riscaldamento si incorrerebbe semplicemente in un inutile spreco di energia. Due esempi di utilizzo della resistenza di pre-riscaldamento vengono mostrati in Figura 4-3 e in Figura 4-4, in cui vengono presentati, sul diagramma psicrometrico, due trattamenti dell'aria, entrambi volti ad innalzare sia temperatura che umidità di quest'ultima.

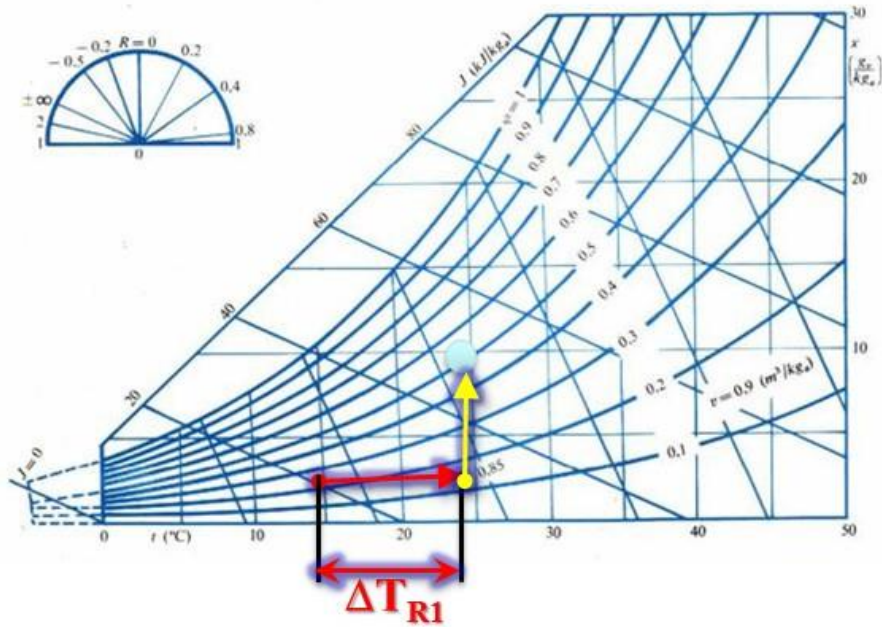


Figura 4-3: Pre-riscaldamento e umidificazione

- La resistenza di post-riscaldamento viene utilizzata quando la temperatura richiesta dell'aria in aspirazione al motore è maggiore del valore di temperatura misurato a valle della batteria di raffreddamento. Ciò può accadere in due casi:
 - quando non è richiesta una diminuzione dell'umidità, per cui la resistenza di pre-riscaldamento è già attiva per scaldare l'aria ma la potenza erogata da essa non è sufficiente a raggiungere il target di temperatura (caso rappresentato in Figura 4-4);

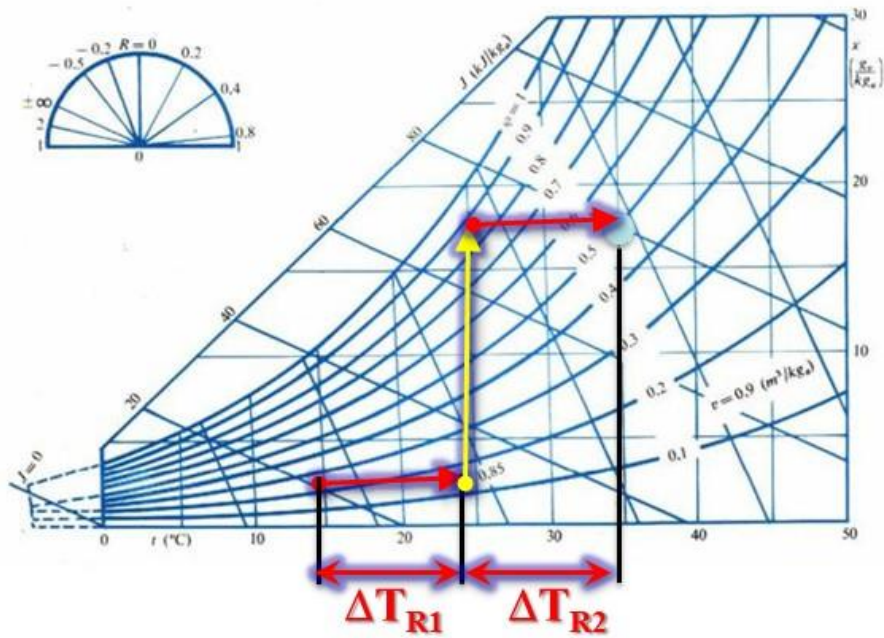


Figura 4-4: Pre-riscaldamento, umidificazione e post-riscaldamento

- quando è necessario deumidificare per raggiungere il target di umidità, per cui la resistenza di post-riscaldamento deve essere attivata per portare la temperatura al valore richiesto, dopo che la batteria di raffreddamento ha portato l'aria al di sotto del punto di rugiada per far condensare il vapore acqueo (caso rappresentato in Figura 4-5).

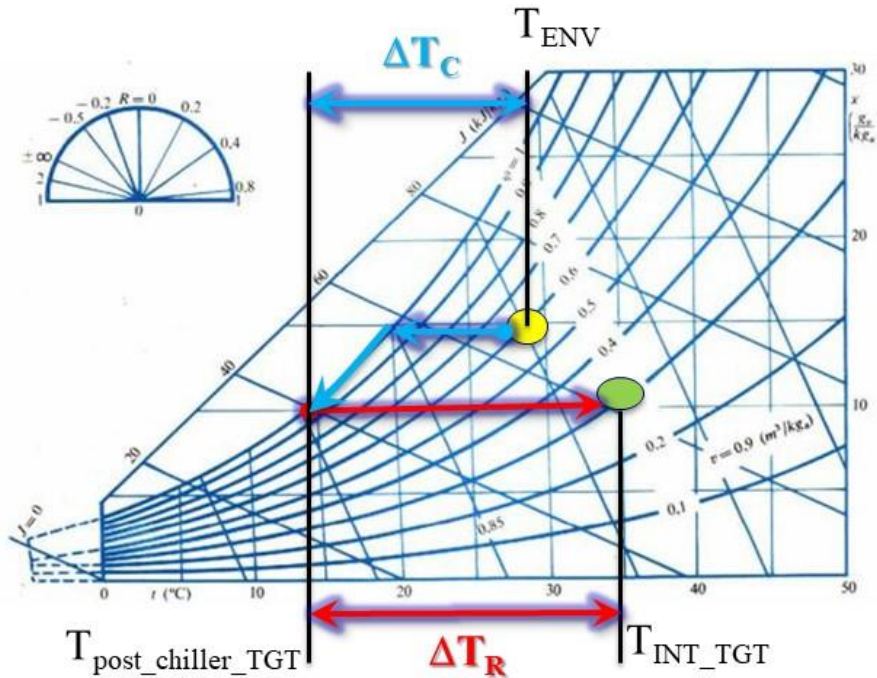


Figura 4-5: Deumidificazione e post-riscaldamento

- Il generatore di vapore viene utilizzato ovviamente solo nel caso in cui sia necessario introdurre vapore acqueo in seno all'aria trattata (casi rappresentati in Figura 4-3, Figura 4-4 e Figura 4-6).
- L'utilizzo della batteria di raffreddamento per raffreddare l'aria, può esser necessario:
 - per rispondere all'esigenza di raggiungere una temperatura in aspirazione motore inferiore a quella di ingresso nell'unità di trattamento aria (caso rappresentato in Figura 4-6);
 - per raggiungere una temperatura dell'aria tale da essere inferiore al punto di rugiada per lo specifico valore di umidità richiesta (caso rappresentato in Figura 4-5).

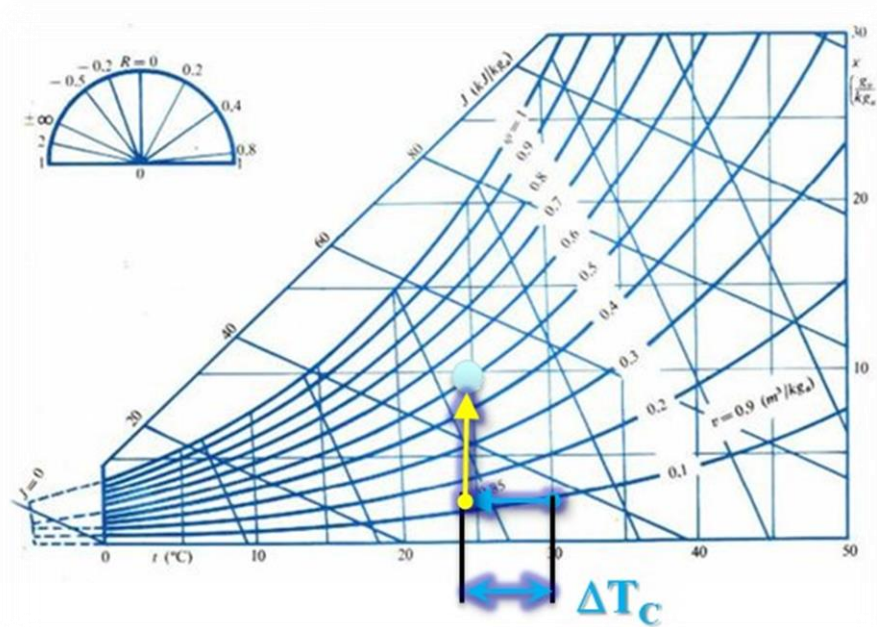


Figura 4-6: Raffreddamento e umidificazione

Per valutare il contributo in catena aperta relativo ad ogni singolo componente è utile suddividere il sistema di condizionamento dell'aria in due sezioni: la prima costituita da resistenza di pre-riscaldamento, batteria di raffreddamento e generatore di vapore, mentre la seconda contenente la sola resistenza di post-riscaldamento. Per gestire i componenti della prima sezione occorre innanzitutto determinare i valori di umidità assoluta richiesta ed effettiva. Il calcolo di queste variabili viene eseguito sfruttando il diagramma psicrometrico, utilizzando come parametri la pressione misurata sul momento dalla stazione barometrica, e umidità relativa e temperatura (valori target nel caso del calcolo dell'umidità assoluta target, valori misurati nel caso dell'umidità assoluta effettiva). Moltiplicando la portata d'aria trattata per la differenza tra i due valori di umidità assoluta (target meno effettiva) appena calcolati si ottiene la portata di vapore acqueo che il sistema deve scambiare con l'aria per raggiungere il valore di umidità relativa desiderato.

$$\dot{m}_{VOL} = \dot{m}_{IA} \cdot (U_{abs_{TGT}} - U_{abs_{ENV}}) \quad 4-2$$

Il contributo fornito dalla retroazione viene utilizzato per correggere direttamente il valore di portata di vapore da scambiare stimato con l'equazione 4-2. In questo caso, il PI utilizzato

viene alimentato con la differenza tra il valore effettivo e quello desiderato di umidità relativa: ciò è dovuto al fatto che le condizioni target di aria comburente sono espresse in termini di umidità relativa e non assoluta. Il fenomeno del wind-up viene gestito dall'algoritmo per evitare che, in caso di saturazione delle attuazioni con errore di regolazione non nullo, il contributo integrale continui a crescere, rendendo poi inattivo il PI. In particolare, la desaturazione del termine integrale tiene conto dell'effetto del guadagno applicato dalla parte di calcolo in catena aperta [17]. Infine, la portata in massa di vapore acqueo da scambiare con l'aria viene calcolata come somma dei due contributi:

$$\dot{m}_V = \dot{m}_{VOL} + \dot{m}_{VCL} \quad 4-3$$

Tale portata può essere negativa o positiva, in base al fatto che l'umidità dell'aria debba essere aumentata o diminuita. Nel primo caso il generatore di vapore verrà attivato per soddisfare la richiesta, viceversa nel caso opposto sarà il chiller ad occuparsi di abbattere la temperatura dell'aria per far condensare il vapore. L'equazione precedente può anche essere scritta per evidenziare i contributi dei due componenti:

$$\dot{m}_{VSG} + \dot{m}_{VCH} = \dot{m}_V \quad 4-4$$

Se $\dot{m}_V > 0 \rightarrow \dot{m}_{VSG} = \dot{m}_V$, al contrario $\dot{m}_V < 0 \rightarrow \dot{m}_{VCH} = \dot{m}_V$: quando è necessario aumentare l'umidità, viene attivato il generatore di vapore mentre il chiller viene spento e viceversa nel caso opposto. In realtà viene mantenuta sempre una minima richiesta di attivazione per entrambe le macchine in modo da evitare i lunghi tempi di riattivazione di queste.

La gestione del generatore di vapore avviene tramite un segnale analogico, che viene calcolato come il rapporto tra la richiesta di portata di vapore e la portata massima che può produrre lo strumento, moltiplicata per un fattore di conversione (f_{SG}) per passare da unità ingegneristiche ad elettriche. L'equazione 4-5 mostra come il prodotto della precedente equazione viene utilizzato per determinare il segnale analogico SG che consente di gestire il generatore di vapore. Per i motivi precedentemente spiegati, come si nota dalla formula, una minima produzione di vapore viene richiesta anche quando non è necessario aumentare

l'umidità dell'aria comburente. In caso contrario, al target di portata di vapore viene sommato un ulteriore contributo per tener conto dell'eventuale diminuzione di umidità, dovuta al fatto che anche il chiller, come vedremo, viene mantenuto in funzione da una richiesta di attivazione minima nel caso in cui non sia necessario deumidificare.

$$SG = \begin{cases} \frac{\dot{m}_V + \dot{m}_{VCHmin}}{SG_C} \cdot f_{SG}, & \dot{m}_V > 0 \\ \frac{\dot{m}_{VSGmin}}{SG_C} \cdot f_{SG}, & \dot{m}_V \leq 0 \end{cases} \quad 4-5$$

Le stesse considerazioni possono essere ripetute per la gestione del chiller. In questo caso il target viene espresso in termini di umidità assoluta a valle della batteria di raffreddamento.

$$Uabs_{CH} = \begin{cases} \frac{\dot{m}_V - \dot{m}_{VSGmin}}{\dot{m}_{IA}} + Uabs_{ENV}, & \dot{m}_V < 0 \\ Uabs_{ENV} - \frac{\dot{m}_{VCHmin}}{\dot{m}_{IA}}, & \dot{m}_V \geq 0 \end{cases} \quad 4-6$$

Utilizzando il diagramma psicrometrico, il valore di umidità assoluta richiesto può essere facilmente convertito a una temperatura target dell'aria a valle dello scambiatore. Dunque, l'attivazione del chiller viene stabilita sommando due contributi, entrambi espressi come un target di temperatura aria a valle della sezione di raffreddamento, uno volto al raggiungimento dell'umidità desiderata, l'altro all'ottenimento della temperatura desiderata. A partire da questo dato, una volta stimata l'efficienza dello scambiatore, si potrebbe calcolare la temperatura del refrigerante necessaria. Purtroppo, tale variabile non può essere attualmente controllata in remoto e viene dunque mantenuta costante. Solo l'attivazione/disattivazione del chiller, tramite un segnale digitale, può esser controllata in remoto. A tale scopo viene utilizzato un segnale PWM, determinato sulla base della potenza termica che il chiller deve estrarre dall'aria. Tale potenza è legata alla variazione di temperatura aria (differenza tra temperatura aria ambientale e a valle della sezione di raffreddamento) che deve essere generata dalla batteria di raffreddamento:

$$\dot{H}_1 = \dot{m}_{IA} \cdot C_{IA} \cdot \Delta T_{EC} \quad 4-7$$

Il duty cycle del segnale PWM viene stabilito come rapporto tra la potenza richiesta per ottenere il calo di temperatura desiderato e la massima capacità di raffreddamento (stabilita per un certo refrigerante utilizzato e per una specifica temperatura ambientale) del chiller.

Quando l'umidità dell'aria deve essere aumentata, il target di temperatura a valle della batteria di raffreddamento viene impostato pari al valore desiderato in aspirazione al motore. Questo è uno dei casi in cui potrebbe accadere che il delta di temperatura, presente nell'equazione 4-7, potrebbe non essere negativo. Quando si verifica questa condizione non è necessaria l'attivazione del chiller (che quindi si limita ad accendersi per rispettare la saturazione sulla richiesta minima), ma l'equazione 4-7 viene sfruttata per determinare il duty cycle del PWM per la gestione della resistenza di pre-riscaldamento. In definitiva gli attuatori vengono controllati nel seguente modo:

$$CH\% = \frac{\dot{H}_1}{CH_{cc}} \quad 4-8$$

$$RpreH\% = \frac{\dot{H}_1}{WRpreH} \quad 4-9$$

L'equazione 4-7 può essere ripetuta per determinare la potenza termica che il sistema di condizionamento deve scambiare con l'aria trattata nella seconda sezione, ovvero quella contenente la sola resistenza di post-riscaldamento.

$$\dot{H}_{2-OL} = \dot{m}_{IA} \cdot C_{IA} \cdot \Delta T_{CR2} \quad 4-10$$

In questa sezione il controllo in catena chiusa consiste in un controllore PI alimentato con la differenza tra la temperatura effettiva in aspirazione motore e quella richiesta. Esso produce un contributo in termini di potenza termica pari a \dot{H}_{2-CL} che viene sommato al contributo in catena aperta per ottenere la potenza totale che la resistenza di post-riscaldamento deve trasferire all'aria comburente.

$$\dot{H}_2 = \dot{H}_{2-OL} + \dot{H}_{2-CL} \quad 4-11$$

Infine, l'attuazione della resistenza di post-riscaldamento può essere scritta come:

$$R_{postH\%} = \frac{\dot{H}_2}{WR_{postH}}$$

La Figura 4-7 è una rappresentazione schematica del modello di controllo del sistema di condizionamento aria comburente.

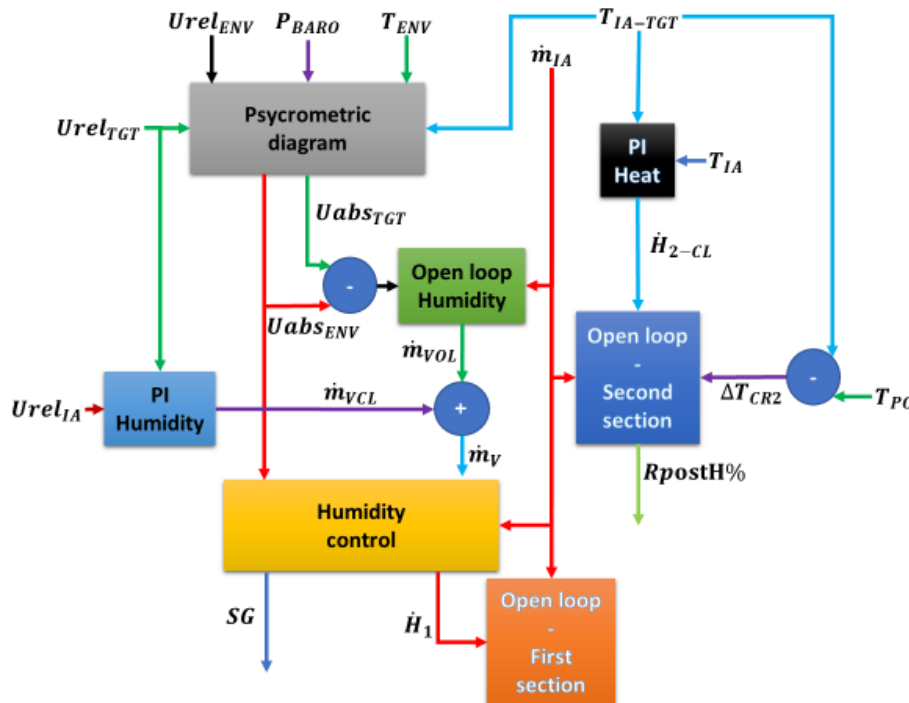


Figura 4-7: Diagramma a blocchi controllo UTA

4.2.2 Controllo temperatura acqua motore

I motori a combustione interna vengono comunemente raffreddati con del liquido [18] (miscela di acqua demineralizzata e glicole etilenico). Durante i test al banco prove, la temperatura del liquido di raffreddamento è uno di quei parametri che viene costantemente monitorato e gestito poiché, influenzando la temperatura delle pareti dei cilindri, ha un effetto non trascurabile sullo sviluppo della combustione. Per questo motivo, durante la fase di sviluppo, i propulsori vengono sottoposti a test con temperature estreme del liquido, per verificare le performance e l'affidabilità dei componenti ad alte temperature, mentre alle basse temperature, dove la combustione potrebbe essere instabile, si monitorano le emissioni.

Per quanto riguarda il caso in questione, essendo la sala prove dedicata a test su motori da corsa, l'attenzione è più rivolta alle temperature elevate. In questa applicazione, come accade solitamente, la temperatura del liquido di raffreddamento viene gestita controllando la portata di acqua proveniente dalla torre evaporativa che, all'interno di uno scambiatore a piastre, riceve calore dal liquido di raffreddamento del motore. In questo caso specifico la regolazione avviene controllando la velocità della pompa che mette in circolazione l'acqua della torre, in altri casi tale pompa lavora sempre a punto fisso e si regola l'afflusso di liquido verso lo scambiatore tramite delle apposite valvole.

La variabile che si desidera controllare in questo caso specifico è la temperatura dell'acqua motore all'uscita da quest'ultimo. Anche per quanto riguarda questo sistema di condizionamento, l'algoritmo di controllo gestisce gli attuatori (in questo caso la sola pompa) con dei comandi che sono il risultato di contributo in catena aperta sommato ad uno in catena chiusa. La porzione calcolata in catena aperta, come mostra l'equazione 4-13, deriva direttamente dalla somma di tre termini:

- La potenza termica necessaria per portare il liquido motore dal valore misurato a quello desiderato;
- La potenza termica che viene trasferita dal motore al liquido di raffreddamento, determinata a partire da una tabella di valori sperimentali che rappresentano la frazione di potenza erogata dal motore che viene trasferita al refrigerante sotto forma di calore;
- La potenza termica che viene dissipata dai tubi del circuito di raffreddamento motore presente in cella: tale contributo può essere determinato a livello sperimentale, valutando, senza circolazione dell'acqua di torre, la caduta di temperatura del refrigerante sul circuito per differenti portate del liquido motore e differenti valori medi della temperatura di tale fluido.

$$\dot{H}_{c-OL} = \dot{q}_c \cdot \rho_c \cdot c_c \cdot (T_{c-TGT} - T_{c-OUT}) + \dot{H}_{E-c} + \dot{H}_{C-DIS} \quad 4-13$$

Da notare che il primo termine nell'equazione è negativo solo quando la temperatura effettiva del refrigerante all'uscita del motore è maggiore di quella desiderata, mentre il secondo termine è sempre negativo, infatti il motore si trova sempre nella condizione di scaldare il liquido, di conseguenza questo contributo è sempre calore che deve essere rimosso dal refrigerante. Al contrario il terzo termine è sempre positivo, in quanto la dissipazione di calore dai tubi del circuito ha ovviamente l'effetto di venire in aiuto all'acqua di torre nel compito di rimuovere calore dal liquido motore.

Per quanto riguarda il contributo in catena chiusa, un regolatore PI corregge il contributo in catena aperta, alimentato da un errore definito come la differenza tra la temperatura del liquido all'uscita motore desiderata (T_{c-TGT}) e quella misurata (T_{c-OUT}). Come già visto per l'unità di trattamento aria comburente, anche in questo caso al PI vengono applicate delle saturazioni dinamiche per tener conto degli effetti del contributo in catena aperta, in modo da assicurare sempre una rapida desaturazione del termine integrale del regolatore. La potenza termica totale che l'acqua di torre deve asportare dal liquido motore è espresso dall'equazione 4-14:

$$\dot{H}_{c-TOT} = \dot{H}_{c-OL} + \dot{H}_{c-CL} \quad 4-14$$

Il risultato di tale equazione genera un'attuazione della pompa solo se negativo: infatti, l'unica azione che può essere intrapresa da questo sistema è quella di rimuovere calore dal fluido motore. Una volta determinata la potenza termica da asportare dal refrigerante, si può calcolare facilmente la portata di acqua di torre necessaria per ottenere lo scambio termico desiderato:

$$DEM_{c-pump} = \frac{-\dot{H}_{c-TOT}}{c_{CTW-c} \cdot \rho_{CTW-c} \cdot (T_{CTW-OUT-c} - T_{CTW-IN})} \quad 4-15$$

Dovendo gestire l'inverter della pompa, è necessario tradurre l'informazione di portata richiesta in velocità desiderata. La relazione tra queste due grandezze può essere determinata o dalla geometria della pompa o attraverso la sua caratteristica: in una macchina volumetrica ideale la relazione tra portata e velocità è lineare a prescindere dalla prevalenza. Al segnale di

controllo della pompa viene applicata una saturazione minima che costringe la macchina ad elaborare almeno una certa quantità di fluido, in modo da evitare surriscaldamento dell'acqua di torre all'interno dello scambiatore a piastre in condizioni di portata bassa o nulla.

La Figura 4-8 rappresenta una schematizzazione a blocchi semplificata del controllo di temperatura acqua motore.

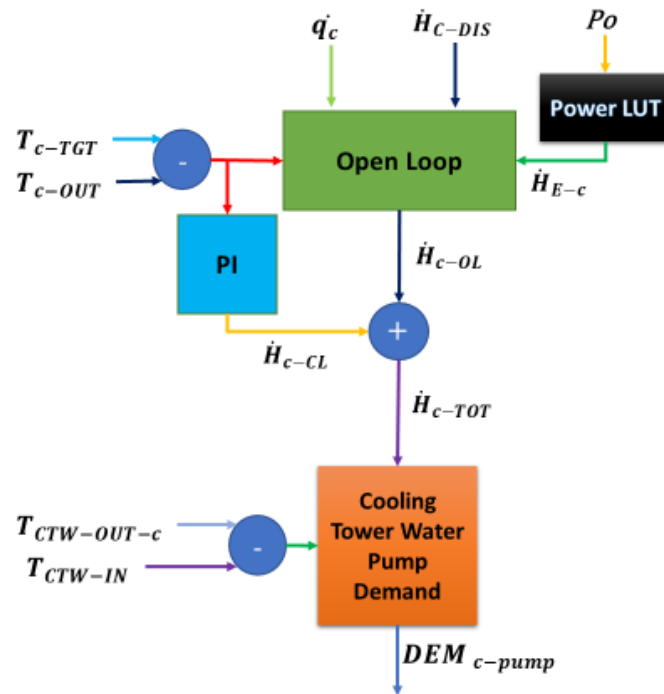


Figura 4-8: Diagramma a blocchi controllo temperatura acqua motore

4.2.3 Controllo temperatura olio motore

I motori a combustione interna necessitano della lubrificazione per ridurre gli attriti che si creano tra le varie parti in movimento, come i pistoni, i cuscinetti, gli alberi, le molle, gli steli delle valvole, ecc. Oltre alla lubrificazione, l'olio motore ha un altro importante compito: rimuovere calore dal motore. Dunque, al fine di evitare malfunzionamenti, rotture e surriscaldamenti del motore, la duplice azione (lubrificante e refrigerante) dell'olio motore è essenziale tanto quanto quella del liquido di raffreddamento.

In maniera del tutto simile a quanto visto per la gestione della temperatura dell'acqua motore, anche il controllo del sistema di condizionamento dell'olio gestisce la portata in massa di acqua di torre che viene inviata ad uno scambiatore a piastre per ridurre la temperatura del lubrificante. Come spiegato in precedenza durante la descrizione dell'hardware dei sistemi, in questo particolare caso di studio, il circuito di condizionamento del lubrificante è equipaggiato con uno scambiatore di calore aggiuntivo, in cui olio diatermico viene utilizzato per scambiare calore con l'olio motore. La temperatura dell'olio diatermico viene gestita da un sistema di condizionamento commerciale dedicato che viene controllato da remoto dal sistema di gestione della sala prove. Il setpoint di temperatura dell'olio diatermico all'ingresso dello scambiatore di calore (ovvero all'uscita del riscaldatore) viene impostato tramite protocollo seriale (RS485). Dunque, in questo caso, il sistema di gestione del banco prove deve gestire due strumenti per controllare la temperatura del lubrificante all'uscita dell'olio motore (variabile controllata).

La gestione della pompa che invia l'acqua di torre allo scambiatore di calore acqua/olio motore avviene con modalità analoga a quanto precedentemente visto nel caso del condizionamento dell'acqua motore: anche in questo caso il contributo in catena aperta è il risultato della somma di tre termini (equazione 4-16).

- la potenza termica necessaria per portare l'olio motore dal valore misurato a quello desiderato;
- la potenza termica che viene trasferita dal motore al lubrificante: i valori sperimentali, che rappresentano la frazione di potenza erogata dal motore che viene trasferita all'olio sotto forma di calore, sono memorizzati in una tabella all'interno del modello;
- la potenza termica che viene dissipata dai tubi del circuito dell'olio motore presente in cella: come nel caso del liquido refrigerante, tale contributo può essere determinato a livello sperimentale, valutando, con la pompa per la circolazione dell'acqua di torre disattivata, la caduta di temperatura dell'olio dall'uscita all'ingresso del motore per differenti portate del lubrificante e diversi valori medi della temperatura di tale fluido.

$$\dot{H}_{O-OL} = \dot{q}_{EO} \cdot \rho_{EO} \cdot c_{EO} \cdot (T_{EO-TGT} - T_{EO-OUT}) + \dot{H}_{E-O} + \dot{H}_{O-DIS} \quad 4-16$$

Il demand di velocità della pompa viene elaborato dal modello tramite l'equazione 4-18, che si basa sulla potenza termica totale che l'acqua di torre deve rimuovere dall'olio motore (equazione 4-17). Il controllo agisce sulla pompa solo se la potenza termica totale risulta essere negativa, ovvero c'è necessità di asportare calore. Ancora una volta viene utilizzato un controllo PI per il contributo in catena chiusa, in questo caso alimentato con la differenza tra valore desiderato ed effettivo della temperatura lubrificante all'uscita del motore. In analogia a quanto visto per l'equazione 4-13, anche per quanto riguarda l'equazione 4-16, il secondo termine è sempre negativo mentre il terzo è sempre positivo.

$$\dot{H}_{O-TOT} = \dot{H}_{O-OL} + \dot{H}_{O-CL} \quad 4-17$$

$$DEM_{O-pump} = \frac{-\dot{H}_{O-TOT}}{c_{CTW-O} \cdot \rho_{CTW-O} \cdot (T_{CTW-OUT-O} - T_{CTW-IN})} \quad 4-18$$

Per quanto riguarda la gestione del sistema di condizionamento dell'olio diatermico, il principio alla base del controllo è ancora una volta quello di valutare la potenza necessaria a raggiungere il valore desiderato di temperatura del lubrificante: in questo sottosistema la variabile controllata è la temperatura dell'olio diatermico all'ingresso dello scambiatore di calore olio diatermico/olio motore. La logica gestione di questo sistema può essere riassunta in tre casi:

- Se la potenza termica totale da trasferire al lubrificante (risultato dell'equazione 4-17) risulta essere positiva, allora la temperatura dell'olio motore va incrementata. In questo caso la pompa che mette in circolazione l'acqua di torre viene fermata (in realtà, come detto in precedenza, c'è una saturazione minima sull'attuazione che costringe la pompa ad elaborare almeno una certa quantità di fluido). Al contempo, il controllo comunica al riscaldatore dell'olio diatermico il risultato dell'equazione 4-19, ovvero il setpoint di temperatura dell'olio diatermico all'uscita del riscaldatore, calcolato sulla base della variazione di temperatura lato olio diatermico stimata per ottenere lo scambio termico desiderato.

$$T_{DO-TGT} = \left(\frac{\dot{H}_{O-TOT}}{\dot{q}_{DO} \cdot C_{DO} \cdot \rho_{DO}} \right) + T_{DO-H-OUT} \quad 4-19$$

- Se invece la potenza termica totale da trasferire al lubrificante è negativa e la pompa dell'acqua di torre non sta ancora elaborando la massima portata possibile, l'algoritmo gestisce lo scambio termico utilizzando la sola pompa. In questo caso il setpoint di temperatura olio diatermico viene impostato al valore di temperatura $T_{EO-C-OUT}$ dell'olio motore all'uscita dello scambiatore acqua di torre/ olio motore. Così facendo si evitano possibili interferenze tra il controllo di temperatura dell'olio diatermico e di quello motore.
- Infine, se \dot{H}_{O-TOT} è ancora una volta negativa ma l'attuazione della pompa è già in saturazione, è necessario ricorrere al sistema di condizionamento dell'olio diatermico per diminuire la temperatura dell'olio motore più rapidamente di quanto farebbe il solo sistema di raffreddamento ad acqua di torre. Il setpoint per la temperatura dell'olio diatermico deriva ancora dall'equazione 4-19, che, questo caso particolare, ha il primo termine a secondo membro negativo, proprio perché è necessario che il controllo richieda un raffreddamento dell'olio diatermico.

Il sistema di condizionamento dell'olio diatermico ha delle dinamiche intrinsecamente lente. Questo è il motivo per cui l'algoritmo di controllo tenta prima di abbassare la temperatura del lubrificante con l'acqua di torre, e poi, se nonostante la pompa sia al massimo delle sue possibilità sia ancora richiesto di raffreddare l'olio motore, anche l'olio diatermico viene utilizzato per il medesimo scopo.

La Figura 4-9 mostra un diagramma a blocchi del controllo sviluppato per la gestione della temperatura dell'olio motore.

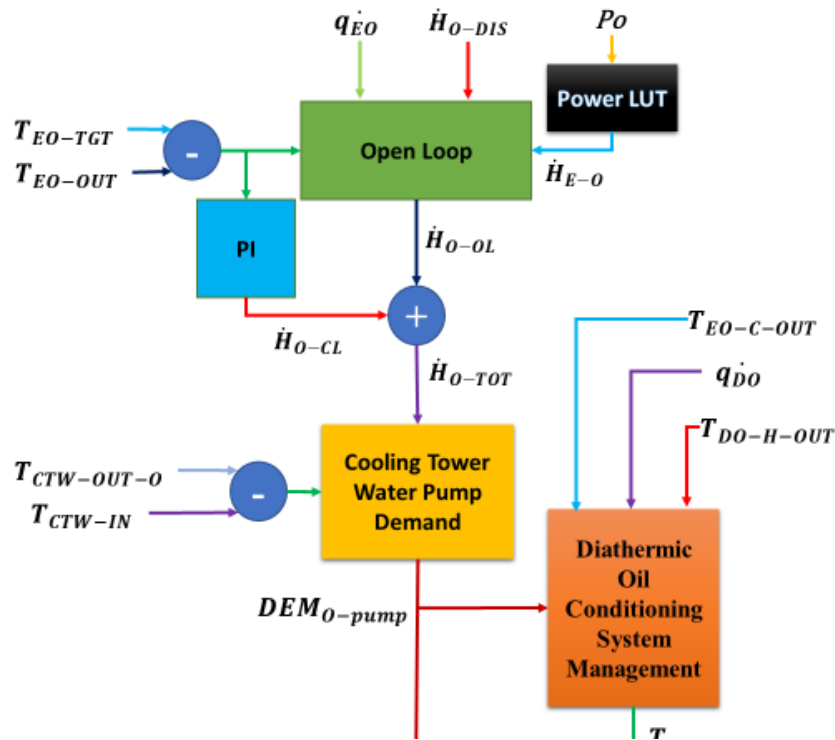


Figura 4-9: Diagramma a blocchi controllo temperatura olio motore

4.3 Risultati

Come già detto in precedenza in questo capitolo, la sala prove in cui è stato validato il modello per la gestione dei sistemi di condizionamento è stata sviluppata e costruita per svolgere test su motori da corsa. Le prove più comuni consistono, per quanto riguarda la calibrazione e la valutazione delle prestazioni, in lenti transitori seguiti da stazionari più o meno lunghi, mentre per valutare l'affidabilità dei componenti del motore e validare le strategie di controllo vengono effettuate delle simulazioni di giri pista (replay di telemetria acquisita sul circuito). La Figura 4-10 mostra come vengono gestiti il regime e il carico del motore durante un tipico test di valutazione delle prestazioni, anche detto curva di potenza. La stessa figura mette in evidenza l'andamento della portata di aria comburente durante la prova: variazioni di flusso così repentine rendono necessario un contributo in catena aperta nella gestione dell'unità di trattamento aria. Analogamente, i transitori di carico e giri motore che si vedono in figura richiedono l'utilizzo di un contributo in catena aperta per quanto riguarda la gestione dei sistemi di condizionamento di acqua e olio motore.

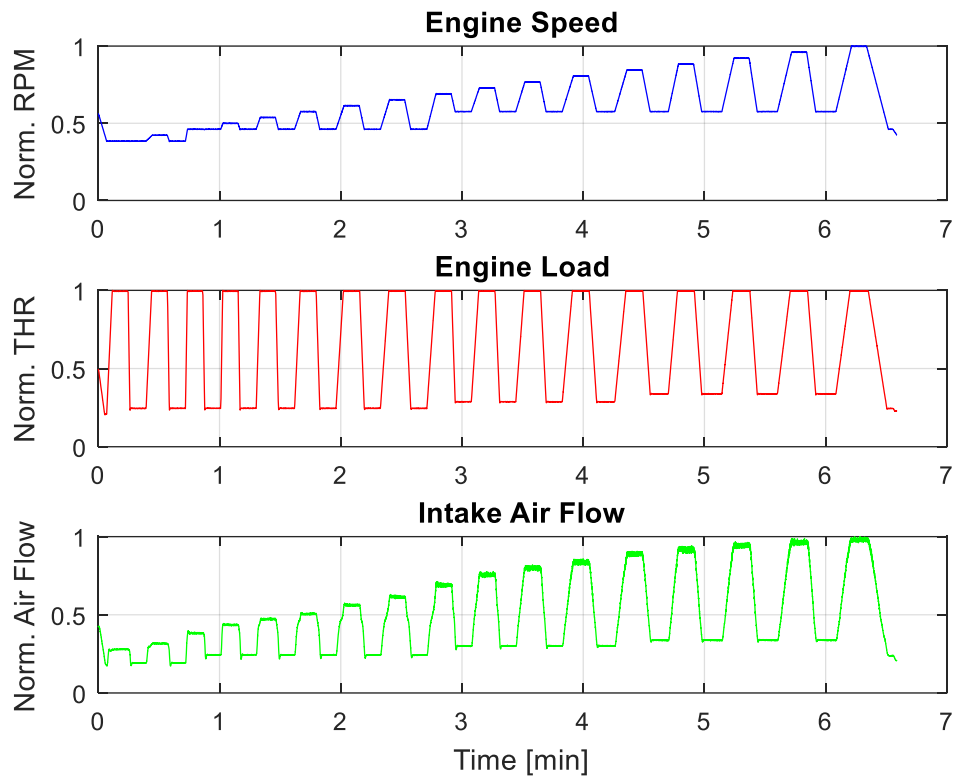


Figura 4-10: Regime motore, carico e flusso d'aria comburente durante una curva di potenza

4.3.1 Calibrazione del controllo

Prima di iniziare la validazione del modello è stato necessario calibrare i parametri del controllo. In questo paragrafo viene brevemente descritta con qualche esempio la calibrazione della sezione di controllo relativa all'unità di trattamento aria. A tale scopo sono state eseguite diverse prove sperimentali con l'unità di trattamento disconnessa dal motore: i parametri del calcolo in catena aperta sono stati ottimizzati disattivando la retroazione, in seguito i guadagni del PI sono stati ottimizzati escludendo questa volta il contributo in catena aperta.

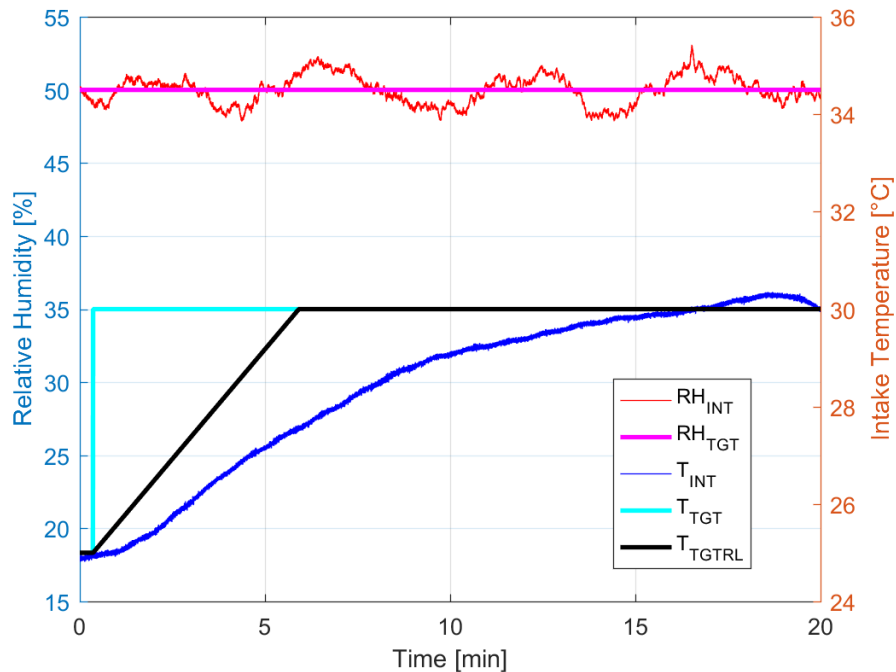


Figura 4-11: Variazioni a gradino della temperatura aria target per la taratura del PI

La Figura 4-11 mostra un esempio di test per l'ottimizzazione dei guadagni del PI per ottenere una corretta risposta dinamica del controllo: vengono richieste delle variazioni a gradino della temperatura aria comburente (da 25 °C a 30 °C) con umidità relativa target costante al 50%. Essendo il contributo in catena aperta disabilitato, la risposta del sistema è lenta, ma l'incremento di temperatura desiderato viene ottenuto senza oscillazioni critiche dell'umidità relativa (rimane all'interno del range $50 \pm 2\%$). La ragione della lentezza nella risposta dipende da diversi fattori:

- come già accennato, la derivata della richiesta viene saturata per evitare di generare istantaneamente un grande errore di regolazione caricando velocemente l'integrale del PI. Dunque, il setpoint di temperatura non cresce a gradino ma in maniera graduale, raggiungendo il target di 30 °C solo dopo 330 secondi;
- le resistenze hanno un periodo di switch che è inevitabilmente lungo e ciò affligge la prontezza del controllo;

- il volume di calma presente al termine del circuito dell'aria va a smorzare le variazioni di temperatura dell'aria comburente.

Il fatto di avere una risposta così lenta è un forte svantaggio nel caso in cui siano richieste delle variazioni repentine delle condizioni dell'aria, tuttavia permette di mantenere tali condizioni stabili durante i transitori di carico e giri motore. Questo secondo aspetto è quello che deve essere privilegiato dato che tipicamente lo svolgimento delle prove al banco prevede di far stabilizzare il sistema raggiungendo le condizioni target dell'aria comburente, che poi vengono mantenute per tutto l'arco del test.

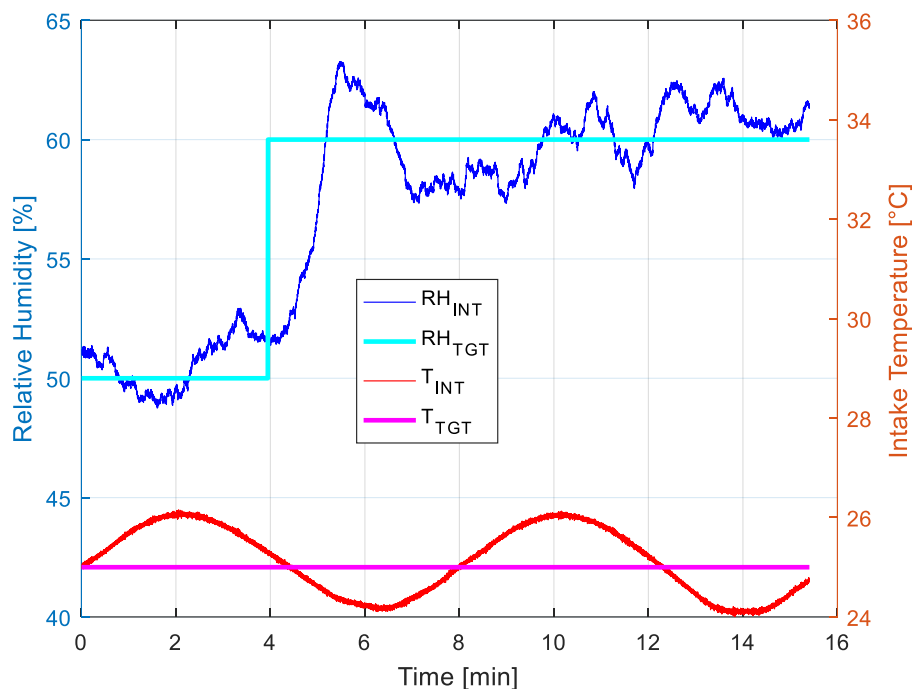


Figura 4-12: Variazioni a gradino dell'umidità relativa per la taratura del PI

La Figura 4-12 mostra la reazione del sistema ad una richiesta di variazione dell'umidità relativa (da 50% a 60%), ad una temperatura dell'aria costante pari a 25 °C: grazie alla corretta calibrazione del PI della temperatura, essa oscilla in un range di ± 1 °C. Le variazioni di temperatura costituiscono un disturbo per il controllo dell'umidità, essendo queste due grandezze tra loro legate. In altri casi si è forzata la variazione di temperatura proprio per mettere alla prova la robustezza del controllo di umidità.

4.3.2 Validazione del controllo

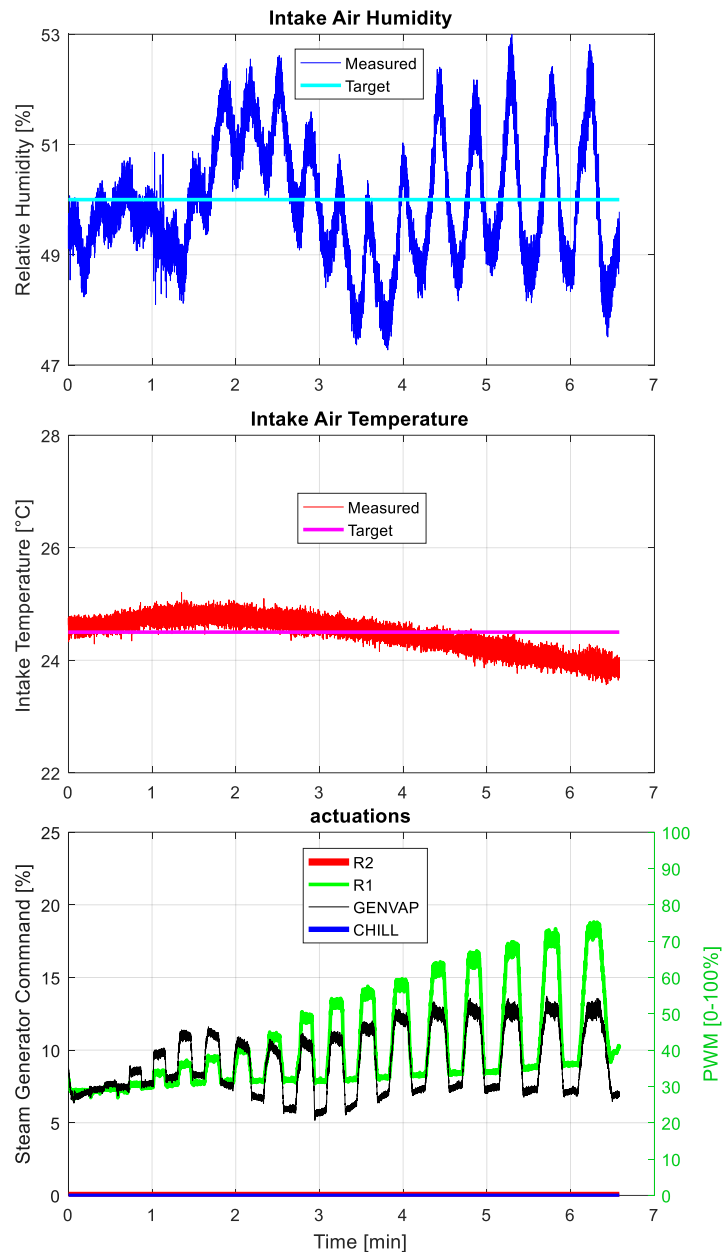


Figura 4-13: Controllo temperatura e umidità aria comburente durante una curva di potenza

La Figura 4-13 mostra le prestazioni e le attuazioni del sistema di condizionamento dell'aria durante un test di valutazione delle prestazioni motore (condizioni operative del motore illustrate in Figura 4-10). Si può osservare che, nonostante il test comporti repentine fluttuazioni della portata d'aria in aspirazione al motore, l'umidità rimane vicina al valore desiderato, con una variazione massima riscontrata durante la prova pari al 3% e una deviazione standard di 1.1%. Dalla figura risulta evidente che durante questo test sia il chiller

che la resistenza di post-riscaldamento erano disattivati, mentre il generatore di vapore e la resistenza di pre-riscaldamento erano attivi: le relative attuazioni sono frutto sia del contributo in catena aperta sia di quello in catena chiusa che cerca di correggere il primo sulla base della differenza tra valore desiderato ed effettivo, come si può notare molto bene dall'attuazione del generatore di vapore a circa due minuti dalla partenza del test.

Per quanto riguarda la gestione della temperatura di acqua e olio motore possono esser fatte considerazioni del tutto simili. La Figura 4-14 mostra che, durante la medesima curva di potenza della precedente figura, il refrigerante viene mantenuto in un range inferiore a $\pm 1^\circ\text{C}$ rispetto al valore desiderato. E' evidente come il contributo in catena aperta sia allineato alle variazioni di regime del motore, grazie al termine \dot{H}_{E-c} dell'equazione 4-13. L'effetto del contributo fornito dalla retroazione si riesce ad osservare solo durante gli stazionari: da notare l'incremento della portata di acqua di torre durante gli ultimi tre stazionari.

La Figura 4-15 mostra il comportamento del controllo della temperatura olio durante lo stesso test delle due figure precedenti. Si può notare che il test comincia prima ancora che l'olio abbia raggiunto il valore di temperatura desiderato. Ciò è dovuto all'implementazione dell'automazione che permette la partenza della curva di potenza quando la temperatura dell'olio è in un range di $\pm 5^\circ\text{C}$ rispetto al target. Ovviamente l'automazione può essere modificata, limitando questo range a beneficio del controllo della temperatura olio. La portata di acqua di torre, non essendo all'inizio della prova necessario raffreddare, è saturata al valore minimo di attivazione della pompa per evitare che nello scambiatore di calore l'acqua della torre cominci a bollire. Nel momento in cui l'olio raggiunge la temperatura desiderata, la pompa comincia a elaborare una portata sempre maggiore, mentre il setpoint di temperatura dell'olio diatermico decresce visibilmente, a causa del fatto che il motore si trova in condizioni operative che lo portano a scaldare l'olio molto velocemente. Escludendo la rampa iniziale, la temperatura dell'olio motore viene mantenuta in un range pari a $[-3; +4]^\circ\text{C}$ rispetto al valore desiderato. L'olio diatermico non viene mai utilizzato in questo caso per il raffreddamento poiché il comando della pompa non arriva mai a saturazione durante l'arco della prova.

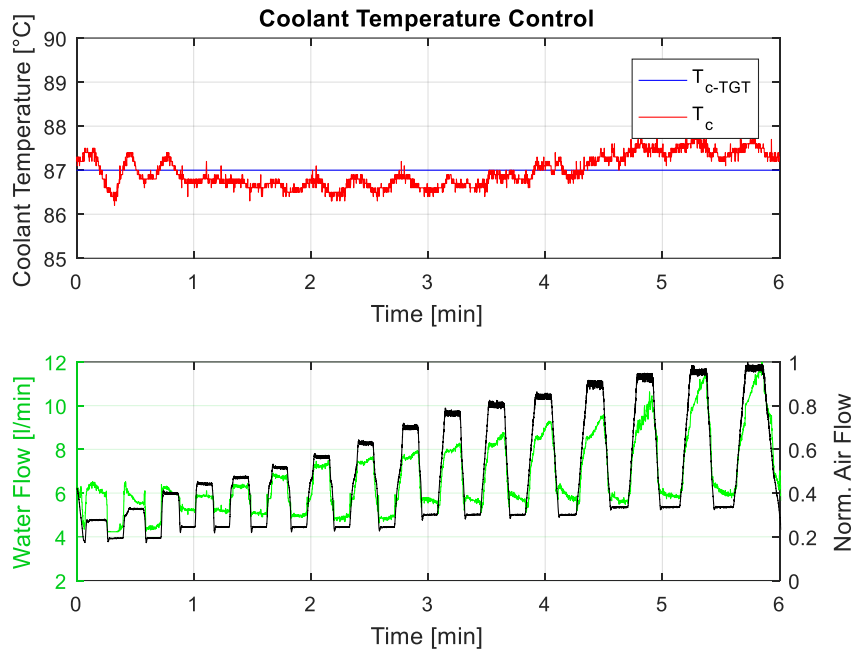


Figura 4-14: Controllo temperatura acqua motore durante una curva di potenza

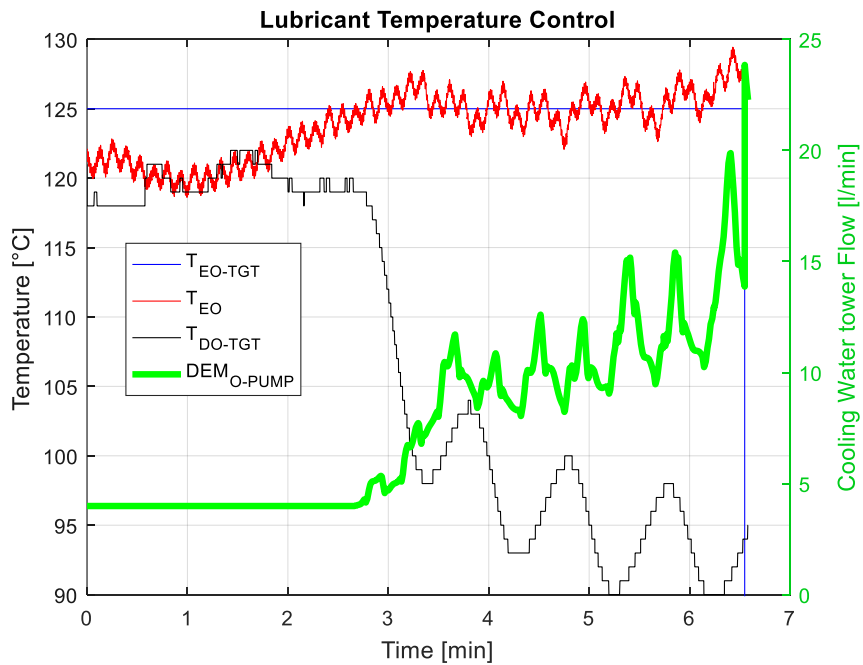


Figura 4-15: Controllo temperatura olio motore durante una curva di potenza

La Figura 4-16 si riferisce ad un test molto diverso da quello visto nelle immagini precedenti: si tratta di una simulazione di un giro pista che viene replicato per un certo numero di volte (la figura mostra solo una porzione del test) per determinare l'affidabilità dei componenti e

validare le strategie di controllo motore implementate in centralina. In queste condizioni i flussi di acqua e olio all'interno del motore sono altamente transitori a causa delle variazioni repentine del regime motore. Nonostante le condizioni altamente variabili, la temperatura dell'acqua motore viene mantenuta in un range di $\pm 2^{\circ}\text{C}$ rispetto al valore desiderato, mentre l'olio oscilla intorno al target di $\pm 3^{\circ}\text{C}$. Questo particolare test viene eseguito senza connettere il motore al sistema di trattamento dell'aria comburente.

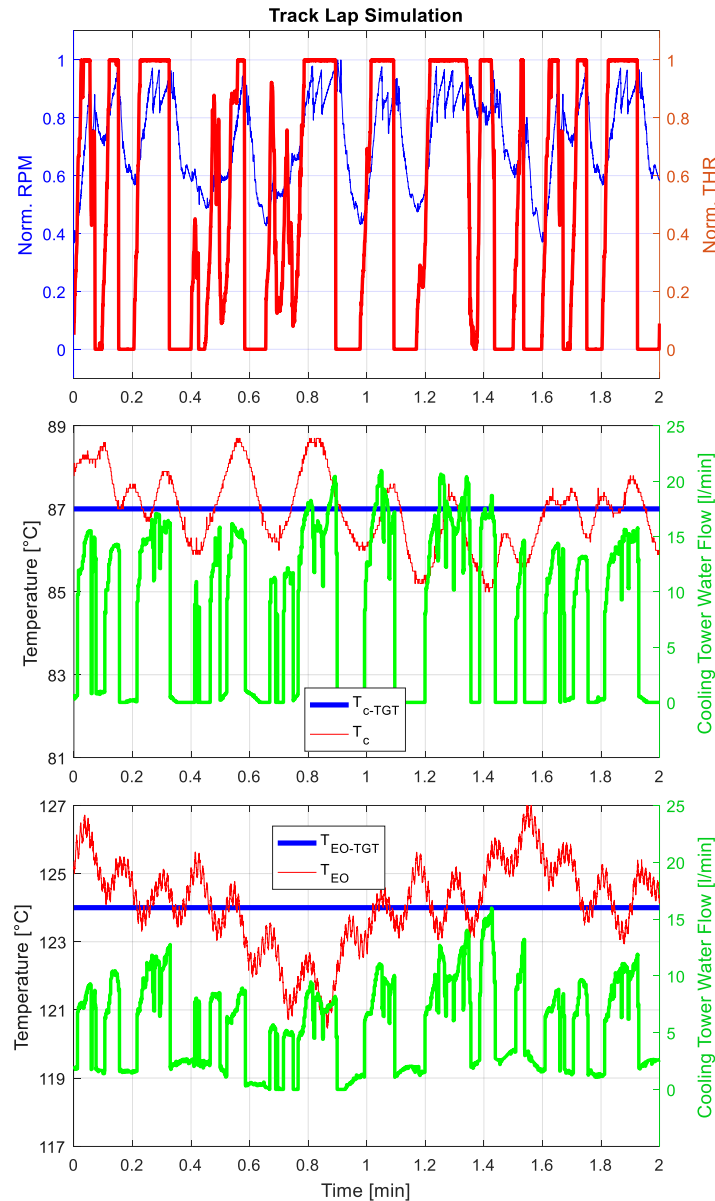


Figura 4-16: Controllo acqua e olio motore durante una simulazione di giro pista

5. Modello termico del sistema di raffreddamento motore

I parametri dei modelli descritti nel quarto capitolo necessitano di essere leggermente ritoccati ogni volta che viene installato un motore diverso, o uguale ma con componenti differenti, che possono alterare i meccanismi di scambio termico intrinseci del propulsore. Un modello in grado di descrivere le dinamiche termiche del motore, al variare dei componenti in esso installati, è uno strumento che permette di validare e calibrare i modelli per la gestione dei sistemi di condizionamento dei fluidi motore [9, 10, 11, 23, 24] (sia sul veicolo che in sala prove) attraverso delle semplici simulazioni, senza la necessità di effettuare queste attività al banco prove con il motore vero. Questa è stata una delle motivazioni principali che ha portato ad affrontare questo tema (un'evoluzione del modello presentato nell'elaborato di tesi di Lorenzo Zoffoli [29]): la modellazione del sistema di raffreddamento di un motore.

Il sistema di raffreddamento gioca un ruolo fondamentale per la sopravvivenza e l'affidabilità del motore, mantenendo in qualsiasi situazione la temperatura del liquido refrigerante entro un certo range desiderato. Tuttavia, tale sistema comporta notevoli svantaggi in termini di aerodinamica del veicolo, un problema di non poco conto soprattutto per quanto riguarda i motocicli [19, 20]. Inoltre, la pompa meccanica del sistema di raffreddamento, che permette di mettere in circolazione il fluido refrigerante, assorbe potenza direttamente dall'albero motore, dunque comporta un aggravio dei consumi non trascurabile ad elevati regimi di rotazione [21]. Queste problematiche vengono affrontate dedicando uno studio molto accurato allo sviluppo dei sistemi di raffreddamento dei motori in modo da minimizzare sia le perdite di efficienza [22], che il peggioramento aerodinamico [20] causato dall'installazione del radiatore frontale. Un modello come quello presentato in questo capitolo può essere di grande aiuto durante lo sviluppo del sistema di raffreddamento, facilitando il corretto dimensionamento dei componenti del sistema per l'installazione di esso sul veicolo.

5.1 Schema dell'impianto

La Figura 5-1 descrive la struttura di un sistema di raffreddamento per motocicli, evidenziando quei componenti la cui modellazione analitica viene argomentata nei successivi paragrafi di questo capitolo.

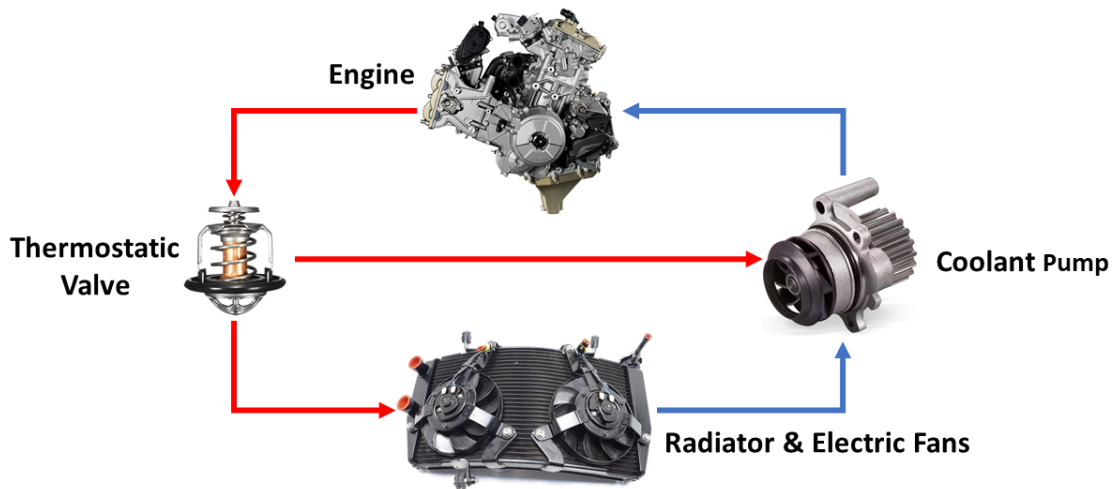


Figura 5-1: Schema del sistema di raffreddamento

Di solito, un sistema di raffreddamento è un insieme di componenti in cui scorre un liquido refrigerante (una miscela di acqua demineralizzata e glicole etilenico) [18]. Come mostrato in Figura 5-1, è costituito da:

- una valvola termostatica, che gestisce il flusso del refrigerante all'uscita del motore: la valvola può consentire o impedire il flusso attraverso il radiatore, a seconda della temperatura del liquido di raffreddamento. Infatti, durante il riscaldamento del motore, la valvola termostatica viene chiusa, evitando il passaggio del refrigerante attraverso il radiatore, in modo da aumentare rapidamente la temperatura del liquido;
- un radiatore, dove avviene lo scambio di calore tra l'aria che impatta frontalmente contro il veicolo in corsa e il liquido di raffreddamento. Di solito il radiatore è dotato di elettroventole che si attivano al di sopra di una certa soglia di temperatura del fluido refrigerante. Accelerando il flusso d'aria attraverso lo scambiatore, le elettroventole garantiscono un corretto raffreddamento del motore anche quando la portata di aria, derivante dalla velocità del veicolo, non è sufficiente a mantenere in target la temperatura del refrigerante (ad esempio a veicolo fermo).
- una pompa, che spinge il liquido di raffreddamento attraverso il circuito. Di solito si tratta di una pompa centrifuga azionata meccanicamente dal motore, ma ultimamente si sta diffondendo l'uso di pompe elettriche, che, essendo svincolate dal propulsore,

possono essere gestite dalla centralina di controllo motore in maniera intelligente con notevoli vantaggi in termini di efficienza rispetto alla soluzione meccanica.

5.2 Modello termico motore

L'obiettivo della sezione di modello relativa al motore consiste nel determinare la variazione di temperatura del refrigerante a cavallo del propulsore (differenza tra ingresso e uscita). Tale differenza può essere valutata mediante un bilancio termico, tenendo in considerazione il calore trasferito alle pareti durante la combustione e la capacità di assorbimento del calore da parte del refrigerante. Le dinamiche termiche sono state implementate facendo un'ipotesi semplificativa: la temperatura del blocco motore è considerata uniforme e pari alla temperatura (T_{c-OUT}) del refrigerante all'uscita dal motore stesso, come se quest'ultimo fosse una superficie di scambio termico infinita. Come mostra l'equazione 5-1, la dinamica del sistema è dominata dalla capacità termica del motore (C_E):

$$\frac{dT_{c-OUT}}{dt} = \frac{1}{C_E} [\dot{H}_{E-c} - \dot{H}_{c-E}] \quad 5-1$$

La potenza termica \dot{H}_{E-c} che il processo di combustione riversa sul refrigerante, viene definita sulla base di dati sperimentali rilevati in condizioni di velocità di rotazione motore e carico costanti, così come la temperatura del refrigerante, fissata a 90 °C. Tali valori rilevati sperimentalmente sono memorizzati all'interno di una mappa 2D.

Il modello non tiene conto direttamente dello scambio termico per convezione tra aria e motore, che dipende dalla temperatura del blocco. Durante l'inizio della fase di riscaldamento, quando il propulsore si trova ad una temperatura simile a quella dell'aria, il suddetto contributo è trascurabile, di conseguenza una porzione maggiore di calore che la combustione ha trasferito alle pareti del cilindro verrà poi ceduto al liquido refrigerante. Viceversa, mano a mano che il motore si riscalda, il contributo dovuto alla convezione aumenta sempre più. Per tenere conto dell'entità dello scambio termico per convezione tra motore e aria, l'output della tabella che riporta i valori di \dot{H}_{E-c} viene corretto con un opportuno termine che dipende dalla

temperatura del blocco motore, che, grazie all'ipotesi semplificativa precedentemente introdotta, si assume essere pari alla temperatura del refrigerante all'uscita del propulsore.

Infine, la potenza termica che il refrigerante riesce effettivamente ad estrarre dal motore può essere calcolata tramite l'equazione 5-2.

$$\dot{H}_{c-E} = \dot{m}_c \cdot C_c \cdot (T_{c-OUT} - T_{c-IN}) = \dot{m}_r \cdot C_c \cdot (T_{c-OUT} - T_{c-OUTr}) \quad 5-2$$

5.3 Modello termico radiatore

Come appena visto nel paragrafo precedente per il motore, anche per quanto riguarda la modellazione del radiatore, il nucleo dell'algoritmo sta nell'equazione differenziale che si basa sul bilancio termico applicato al componente. La stessa ipotesi semplificativa viene applicata anche in questo caso: la temperatura del radiatore viene considerata uniforme e pari a quella del fluido refrigerante in uscita allo scambiatore di calore stesso.

$$\frac{dT_{c-OUTr}}{dt} = \frac{1}{C_r} \cdot [H_{c-r} - H_{r-a}] \quad 5-3$$

La potenza introdotta nel radiatore dal liquido refrigerante può essere calcolata tramite l'equazione 5-4 sfruttando le temperature prodotte dal modello all'iterazione precedente.

$$\dot{H}_{c-r} = \dot{m}_r \cdot C_c \cdot (T_{c-OUTr} - T_{c-INr}) \quad 5-4$$

Lo scambio termico che avviene all'interno del radiatore viene calcolato attraverso il metodo epsilon-NTU [27], utilizzando le sole temperature in ingresso di aria e liquido refrigerante. La capacità termica oraria (\dot{C}), definita come il prodotto tra la portata in massa e il calore specifico, può essere calcolata sia per il ramo freddo che per quello caldo, di conseguenza può essere facilmente determinato anche il rapporto tra la capacità termica oraria maggiore e quella minore. Quale dei due fluidi abbia capacità termica maggiore dipende dalle specifiche condizioni operative di motore e veicolo. L'efficienza di scambio termico può infine essere calcolata come funzione di $\dot{C}_{min}/\dot{C}_{max}$ e $NTU = \frac{u \cdot A}{\dot{C}_{min}}$ utilizzando specifiche tabelle per

radiatori automotive [27]. Per semplicità, nella valutazione del coefficiente di scambio termico globale si assume l'ipotesi di una relazione lineare tra il prodotto $u \cdot A$ e la capacità termica minore.

$$u \cdot A = k1 \cdot \dot{C}_{min} + k0$$

5-5

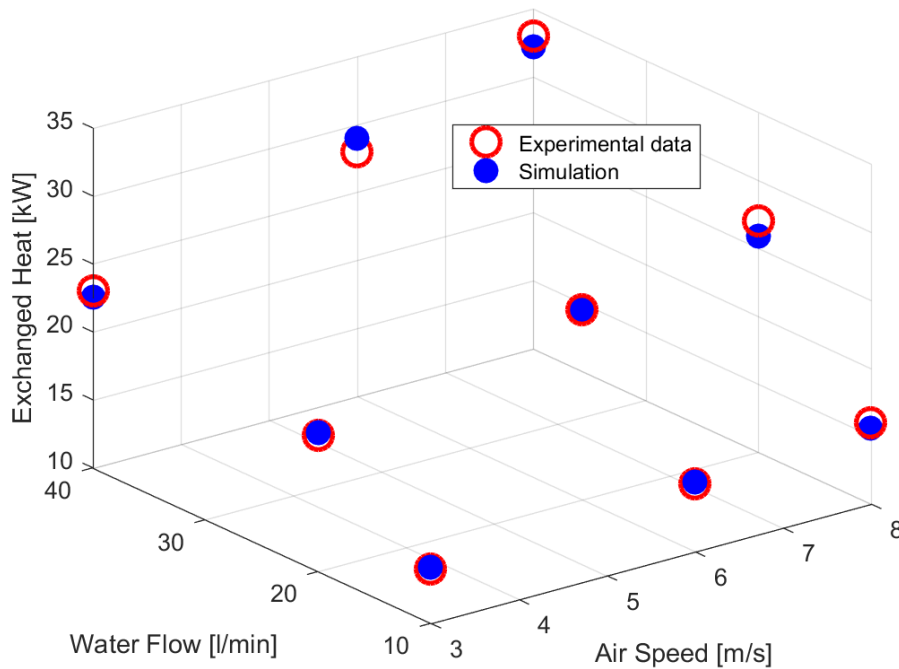


Figura 5-2: Potenza ceduta dal radiatore all'aria, confronto tra dati sperimentali e risultati del modello

I coefficienti $k0$ e $k1$ dell'equazione 5-5 sono stati determinati sulla base di dati sperimentali relativi al radiatore. I test in questione sono stati condotti facendo variare la portata di liquido al radiatore e la velocità dell'aria, misurando all'ingresso e all'uscita dello scambiatore la temperatura del refrigerante, in modo da poter valutare il calore scambiato. Poiché grazie a questi risultati possono essere facilmente valutate le capacità termiche orarie, può essere determinato anche NTU e di conseguenza il prodotto $u \cdot A$, consentendo in questo modo di ottimizzare i coefficienti dell'equazione 5-5. La Figura 5-2 mostra il confronto tra i dati

sperimentali, utilizzati per la calibrazione dei coefficienti, e i risultati prodotti dal modello. La potenza che l'aria estrae dal radiatore può essere infine calcolata attraverso l'equazione 5-6, in accordo con il metodo epsilon-NTU:

$$\dot{H}_{r-a} = \varepsilon \cdot \dot{H}_{r-aMAX} = \varepsilon \cdot \dot{C}_{min} (T_{c-INr} - T_{Ar}) \quad 5-6$$

5.4 Calcolo delle portate

Per svolgere i calcoli illustrati nei due paragrafi precedenti, l'algoritmo deve aver precedentemente determinato la portata di liquido refrigerante attraverso i rami del sistema (motore e radiatore) e la portata di aria attraverso il radiatore in diverse condizioni operative. In questa sezione viene illustrato come i vari componenti del circuito sono stati modellati per consentire la determinazione delle portate.

La caratteristica della pompa è stata modellata facendo un "fitting" di dati sperimentali, che riportavano prevalenza Δp e portata Q elaborata dalla pompa in differenti condizioni di velocità motore n , con una espressione polinomiale di terzo grado. In accordo con la teoria della similitudine fluidodinamica [25], i parametri operativi sono stati normalizzati rispetto ad una velocità motore di riferimento, come indicato dall'equazione 5-7, in modo da ridurre le varie curve relative a diverse velocità ad una caratteristica unica, espressa dall'equazione 5-8.

$$n_{red} = \frac{n}{n_{ref}}; Q_{red} = \frac{Q}{Q_{ref}}; \Delta p_{red} = \frac{\Delta p}{\Delta p_{ref}} \quad 5-7$$

$$\Delta p_{red} = a \cdot Q_{red}^3 + b \cdot Q_{red}^2 + c \cdot Q_{red} + d \quad 5-8$$

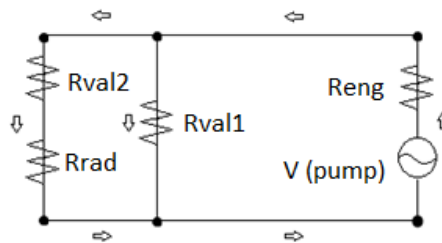


Figura 5-3: Analogia elettrica del circuito di raffreddamento

Come mostra la Figura 5-3, il circuito idraulico di raffreddamento può essere facilmente modellato ricorrendo all'analogia elettrica [26]. Per descrivere la relazione tra portata e differenza di pressione di ogni componente del circuito (motore, radiatore e valvola termostatica) si utilizza l'espressione $\Delta p = K \cdot Q^2$, dove K è la resistenza idraulica, analoga della resistenza elettrica R . Basandosi sui dati sperimentali, le resistenze relative al motore e al radiatore possono essere considerate costanti. Per quanto riguarda la valvola termostatica, il suo comportamento viene descritto nel modello tramite due resistenze idrauliche, una sul ramo del radiatore, l'altra sul ramo di by-pass, variabili e inversamente proporzionali tra di loro. La caratteristica di queste due resistenze dipende dalla temperatura del liquido refrigerante all'uscita del motore, come descritto dalla Figura 5-4

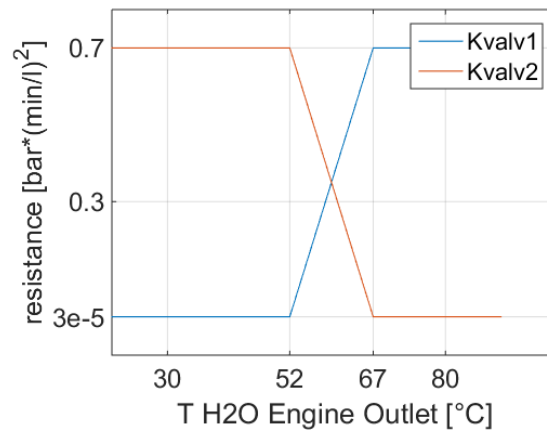


Figura 5-4: Caratteristiche delle due resistenze elettriche equivalenti alla valvola termostatica

Il valore minimo delle due resistenze è stato determinato in modo tale da assicurare su entrambi i rami almeno un trafilemento pari all'1% della portata totale che circola nel circuito. Come mostrato in Figura 5-3 le resistenze R_{val2} e R_{rad} sono in serie fra di loro e in parallelo con R_{val1} . La resistenza equivalente è a sua volta in serie con R_{eng} . Dunque, la resistenza totale del circuito può essere facilmente calcolata attraverso l'espressione 5-9, sfruttando l'analogia tra resistenza elettrica e idraulica.

$$K_{tot} = \frac{K_{V1} \cdot (K_{V2} + K_R)}{K_{V1} + K_{V2} + K_R} + K_E \quad 5-9$$

La condizione di funzionamento della pompa può essere trovata intersecando la curva caratteristica della macchina con la resistenza totale del circuito idraulico. Ciò permette di determinare la portata ridotta (Q_{red}) che attraversa la pompa e, grazie alla similitudine fluidodinamica, è possibile stabilire i valori di portata Q_P e prevalenza Δp_P relativi ad uno specifico valore di velocità di rotazione del motore a partire dalle grandezze ridotte:

$$\Delta p_{red} - K_{tot} \cdot Q_{red}^2 = 0; \quad Q_P = Q_{red} \cdot n_{red}; \quad \Delta p_P = \Delta p_{red} \cdot n_{red}^2 \quad 5-10$$

Le portate di fluido refrigerante che circolano rispettivamente attraverso il radiatore e sul ramo di by-pass possono essere determinate utilizzando l'equazione (5-12), in accordo con l'analogia elettrica:

$$\Delta p_r = (K_{V2} + K_r) \cdot \dot{q}_r^2 = \Delta p_p - K_E \cdot Q_p^2 = \Delta p_{bp} = K_{V1} \cdot Q_{bp}^2 \quad 5-11$$

Per stabilire la capacità termica lato aria è necessario stabilire prima la portata di aria che attraversa il radiatore. Il moto del veicolo genera una portata di aria sullo scambiatore che può essere calcolata attraverso l'equazione 5-12, dove ϵ_{rad} è un coefficiente riduttivo che tiene conto della diminuzione di velocità di impatto dell'aria sul radiatore dovuta alla carenatura del veicolo.

$$\dot{m}_{air\ sp} = sp_V \cdot \rho_{air} \cdot A_{rad} \cdot \epsilon_{rad} \quad 5-12$$

Nel momento in cui le elettro-ventole vengono attivate viene aggiunto un ulteriore contributo alla portata totale: la dinamica delle ventole durante i transitori di velocità viene simulata con una funzione di trasferimento del primo ordine, mentre l'impatto che il vento generato dal moto del veicolo potrebbe avere sul flusso provocato dalle elettro-ventole viene trascurato. All'interno del modello l'azionamento delle elettro-ventole viene gestito, come del resto in centralina, con una semplice isteresi.

Per rappresentare il trasporto di massa dal motore al radiatore e viceversa, il codice prevede dei blocchi di ritardo temporale: il ritardo è funzione della portata di refrigerante.

5.5 Risultati delle simulazioni

Il modello è stato calibrato utilizzando i risultati di due prove sperimentali, una in condizioni di motore al minimo, l'altra con veicolo lanciato ad alta velocità, presentate rispettivamente in Figura 5-5 e Figura 5-6. I dati relativi al test eseguito al minimo sono stati utilizzati per calibrare le capacità termiche di motore e radiatore e la resistenza idraulica della valvola termostatica, mentre i risultati del secondo test sono stati utili per calibrare l'efficienza aerodinamica del radiatore.

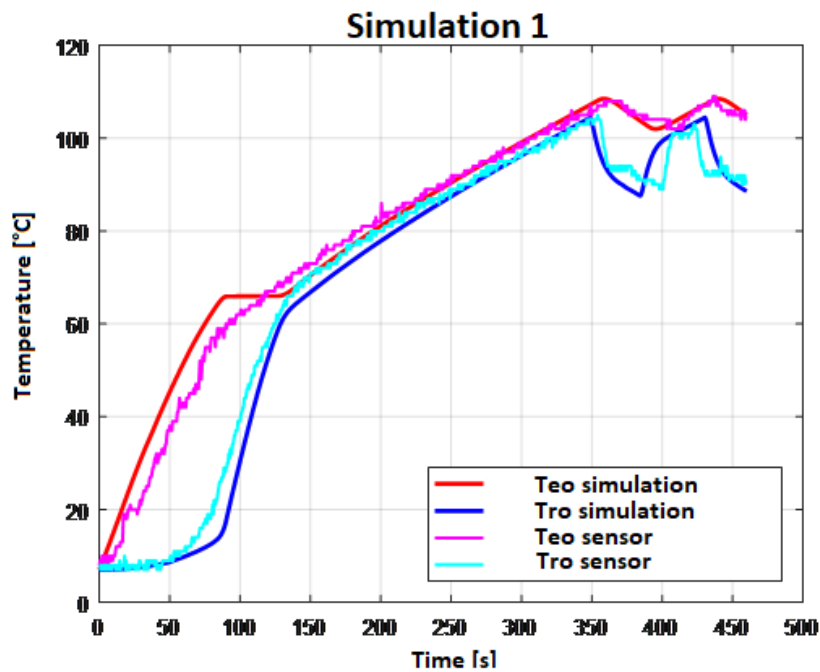


Figura 5-5: Test di riscaldamento con motore al minimo

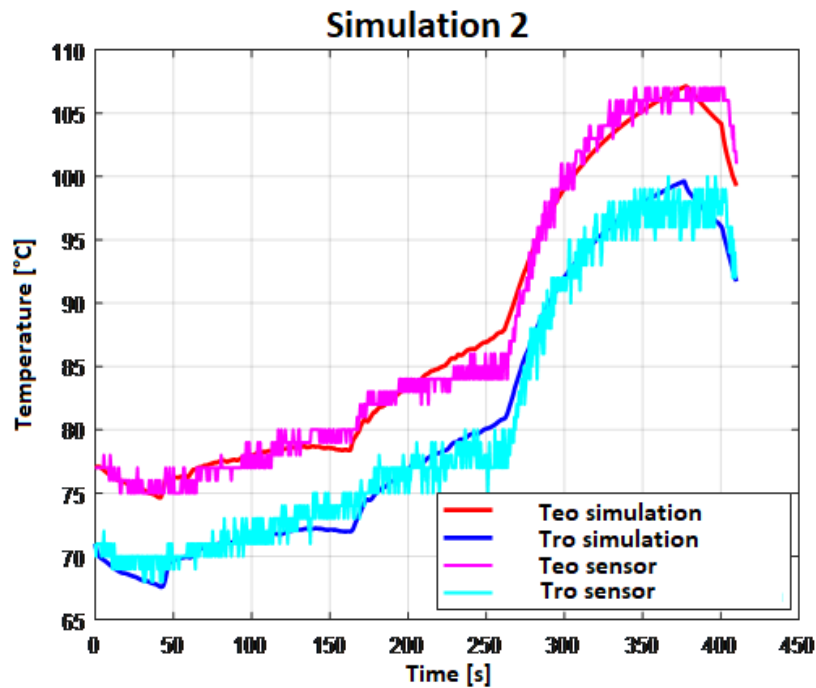


Figura 5-6: Test con veicolo ad alta velocità

Entrambe le figure mostrano il confronto tra dati sperimentali e risultati delle simulazioni. L'effetto della portata di aria prodotta dalle elettro-ventole viene simulato correttamente dal modello come si può riscontrare dalle oscillazioni di temperatura al termine del primo test.

Una volta calibrato il modello, sono state eseguite altre simulazioni. La Figura 5-8 mostra i risultati ottenuti da una simulazione di un giro di pista (le condizioni, in termini di velocità motore, apertura farfalla e velocità veicolo, durante il giro di pista sono riportate in Figura 5-7): il modello riesce a simulare correttamente sia il valore medio che le variazioni della temperatura del liquido di raffreddamento.

La Figura 5-9 mostra un esempio di come il modello possa essere utilizzato per confrontare sistemi di raffreddamento diversi, in termini di influenza sulla temperatura del refrigerante. Nello specifico vengono riportati i risultati di un confronto tra due sistemi, uno con pompa meccanica e l'altro con pompa elettrica, la cui velocità viene regolata con un controllo in catena chiusa alimentato dall'errore tra il target di temperatura del refrigerante all'uscita del motore e il relativo valore misurato. Nelle simulazioni la pompa elettrica è stata controllata con due diversi valori del target di temperatura: nel primo caso si è scelto come obiettivo un valore pari alla temperatura media ottenuta durante la simulazione con pompa meccanica (92

°C); nel secondo caso è stato scelto un target lievemente maggiore (94 °C). La gestione della temperatura del refrigerante tramite il controllo della velocità della pompa porta a dei benefici evidenti, specialmente nel caso in cui il target è maggiore: la minor portata e prevalenza richieste comportano una drastica diminuzione dell'assorbimento di potenza da parte della pompa. Tuttavia, volendo fare un vero e proprio confronto in termini di potenza, è necessario tenere in considerazione l'efficienza della pompa elettrica nei confronti di quella meccanica, l'efficienza dell'alternatore (notoriamente scarsa specialmente nel caso di veicoli con alimentazione elettrica a 12V) e l'efficienza di carica/scarica della batteria. Inoltre, se si pensa ad applicazioni in ambito motociclistico, la pompa elettrica non è una strada facilmente percorribile a causa di questioni legate al peso e agli ingombri.

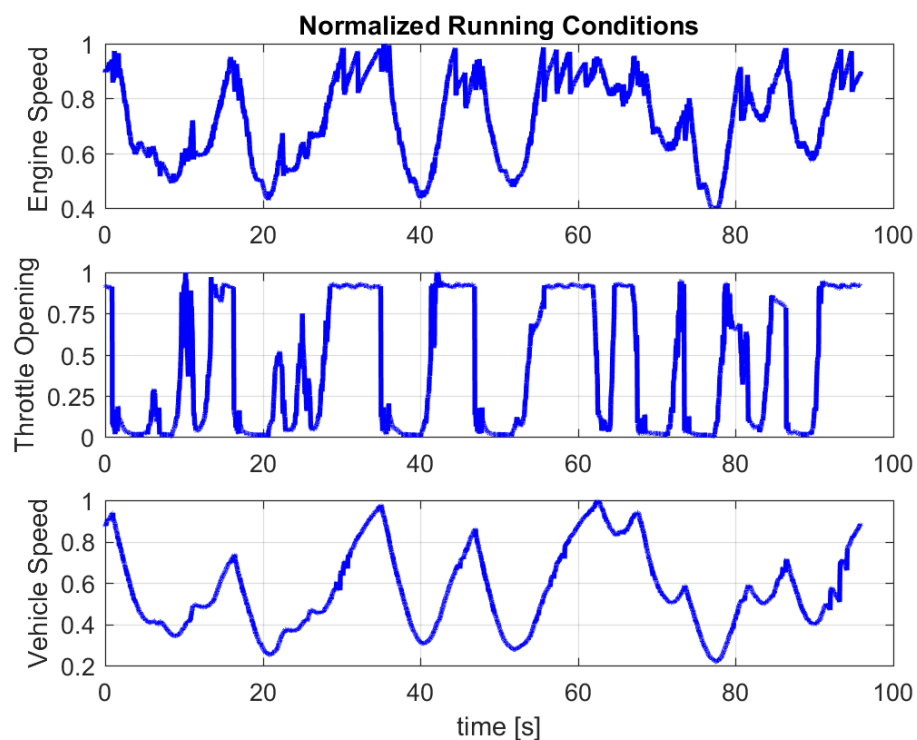


Figura 5-7: Condizioni giro pista normalizzate

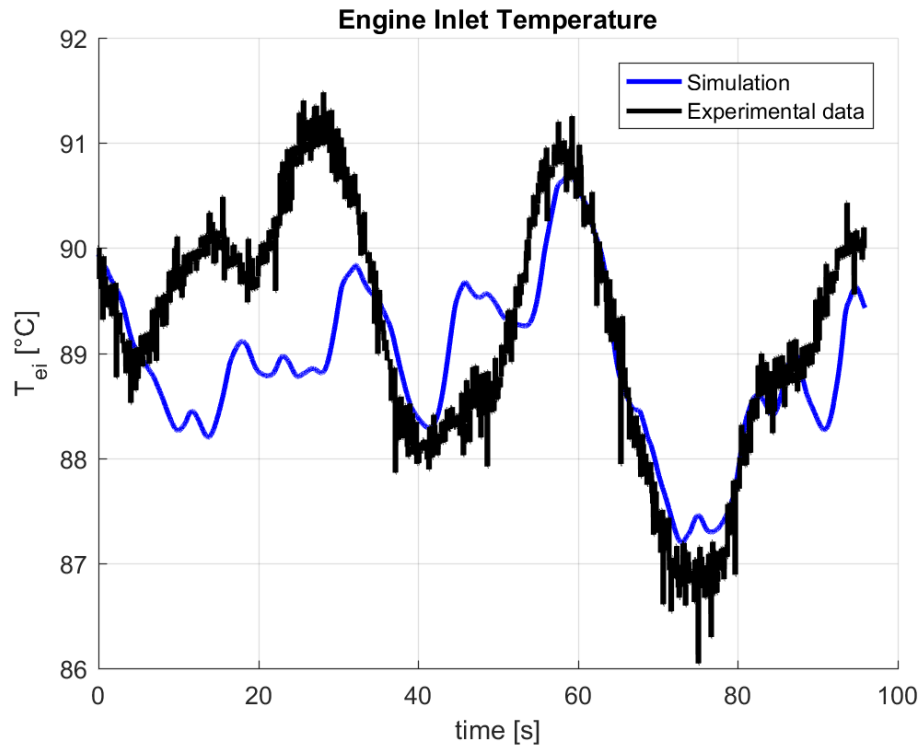


Figura 5-8: Simulazione giro pista

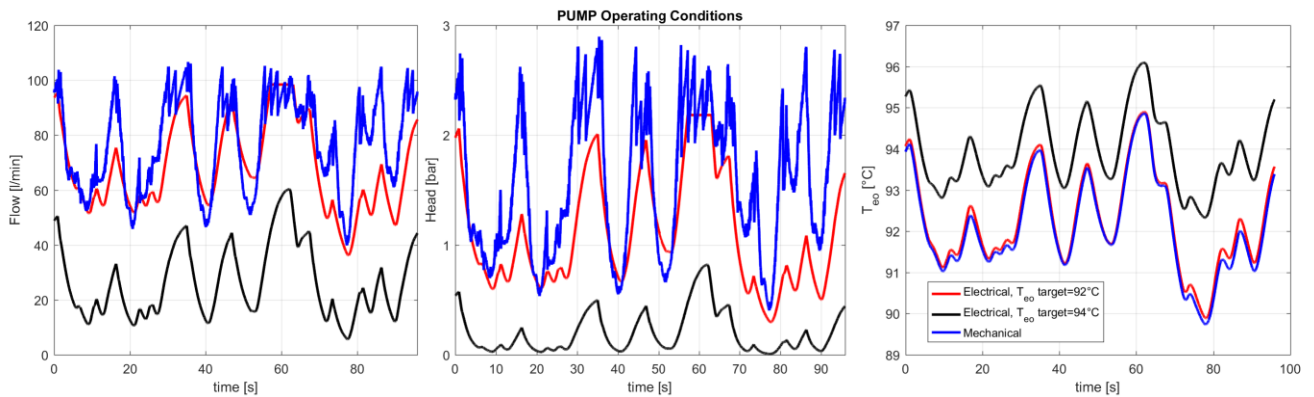


Figura 5-9: (a) Confronto portate, (b) Confronto prevalenze, (c) Confronto temperature refrigerante

In conclusione, i risultati delle simulazioni confermano la validità del modello nel simulare correttamente i meccanismi di scambio termico che avvengono nel sistema di raffreddamento del motore. Un modello simile può essere sviluppato per il motore installato al banco prove e, una volta calibrato e validato, può esser utilizzato per simulare il sistema di raffreddamento del motore permettendo un più rapido sviluppo e validazione dei modelli implementati per la gestione dei sistemi di condizionamento cella che controllano le condizioni dei fluidi motore.

6. L'automazione della sala prove

L'automazione è uno dei temi fondamentali della sala prove. Come già sottolineato nell'introduzione, non è umanamente possibile per un operatore gestire in toto lo svolgimento di una prova al banco come una calibrazione, sia per motivi di ripetibilità, sia per il numero molto elevato di parametri che devono essere monitorati e gestiti con precisione, al fine di ottenere una corretta e accurata esecuzione della prova. Solitamente, all'operatore di cella viene richiesto semplicemente di supervisionare il test, assicurandosi che il motore e l'equipaggiamento del banco lavorino correttamente, e, nel caso in cui si manifesti una situazione di pericolo per i componenti o gli addetti, di agire rapidamente interrompendo il corso della prova, qualora non sia già stato fatto dal sistema di gestione della sala prove.

I cicli automatici vengono utilizzati per controllare il motore, gestendo i demand di carico (acceleratore) e velocità di rotazione del freno (e quindi del motore), durante l'esecuzione di manovre, in alcuni casi semplici, in altri anche complesse, utili per:

- Rilevare le prestazioni del motore: per determinare le performance di un propulsore è pratica comune effettuare la cosiddetta *curva di potenza*, che comporta l'esecuzione di un certo numero di punti in stazionario, ciascuno ad un diverso regime, tutti a pieno carico. Questi stazionari di interesse vengono ovviamente intervallati da stazionari a regime e carico di riposo per evitare di stressare troppo il motore durante il test. Ciò che si ottiene è il rilevamento delle prestazioni in una serie di punti di funzionamento, che nell'insieme costituiscono la caratteristica del motore, in termini di potenza, coppia e consumo specifico: prestazioni fondamentali del motore, che possono essere confrontate con dati storici, per dare indicazioni sull'efficacia dello sviluppo effettuato sul propulsore. Ad esempio, durante la fase di sviluppo, la scelta di montare un componente piuttosto che un altro al fine di aumentarne le prestazioni, viene fatta proprio sulla base dei risultati derivanti dalla curva di potenza. La Figura 4-10 mostra una curva di potenza in termini di regime di rotazione, carico e portata d'aria comburente aspirata dal motore. Il grafico non mostra la precedente fase di riscaldamento, che prevede lo stazionamento in diversi punti di funzionamento per cercare di riscaldare velocemente i liquidi motore spostandosi, mano a mano che la

temperatura dei fluidi cresce, in punti a potenza sempre più alta. Analogamente al termine del test è prevista una fase di raffreddamento, che consente di spegnere il motore sempre ad certa temperatura di riferimento, alla quale è previsto il controllo del livello olio da parte dell'operatore. Anche le fasi di raffreddamento e riscaldamento possono essere gestite in maniera automatizzata per assicurare ripetibilità anche durante queste operazioni, apparentemente banali.

- Determinare l'affidabilità dei componenti: questa attività prevede lo svolgimento di una prova di durata, che consiste nella ripetizione di un ciclo, che rispecchia le condizioni di funzionamento in cui si trova a lavorare il motore durante il suo normale utilizzo, per un numero considerevole di volte. Nel caso di un motore da corsa, essendo sviluppato per essere impiegato su di un circuito, la prova di durata comporta la simulazione delle condizioni riscontrate durante un giro di pista, al banco prova. In questo caso, solitamente, si hanno a disposizione dei dati acquisiti dai sistemi di telemetria, che riportano l'andamento di carico, giri motore ed altri segnali durante il giro sul tracciato. Per far lavorare al banco prove il motore in queste condizioni è necessario riprodurre i dati di telemetria e utilizzare questi valori per controllare i demand di carico e giri motore. Inoltre, la centralina deve essere alimentata con quei segnali analogici, provenienti dai vari sensori che equipaggiano il veicolo, che vengono utilizzati dalle strategie attive durante il giro di pista (ad esempio, velocità ruota anteriore e posteriore, i giroscopi delle IMU (inertial measurement unit), schiacciamento forcella anteriore, pressione ai freni, forza rilevata dalla cella di carico sul cambio, ecc..). E' ovviamente un test molto critico che mette a dura prova i componenti del motore, sia per gli stress provocati da un utilizzo così intenso, sia per la durata di tale prova che può andare avanti per ore (ben oltre la durata di un'ipotetica gara): l'obiettivo è accumulare il maggior numero di chilometri possibile fino alla rottura di un componente. Spesso la simulazione del giro di pista viene utilizzata non solo per determinare l'affidabilità dei componenti ma anche per validare le strategie di controllo motore implementate in centralina.
- Eseguire calibrazione e sviluppo motore: a differenza delle altre tipologie di attività appena descritte, in questo caso non è sufficiente gestire solo i segnali di carico e giri

motore o degli analogici con cui viene alimentata la centralina, nel caso specifico del test di durata. Infatti, durante questi cicli è necessario controllare tutti quei parametri scritti in centralina che richiedono una specifica messa a punto per far sì che le strategie di controllo motore lavorino correttamente, in qualsiasi condizione di funzionamento si trovi il propulsore. Si ricorda che la gestione dei parametri di centralina dall'esterno, attraverso ad esempio protocollo CAN, è una funzionalità che non è prevista per centraline che equipaggiano di serie i veicoli sul mercato. Dunque per questo genere di prove è necessario disporre di una centralina “aperta” di sviluppo, come nel nostro caso specifico: essendo il banco prove oggetto di questa attività concepito per lo sviluppo di motori da corsa, la centralina con cui ci si interfaccia ha al suo interno un software che prevede la possibilità di sovrascrivere alcuni parametri di controllo motore dall'esterno, attraverso protocollo di comunicazione CAN.

6.1 Le sequenze Real-Time

NI Veristand offre all'utilizzatore la possibilità di eseguire delle sequenze di comandi, preconfigurate attraverso un apposito editor, detto Stimulus Profile, rilasciato da NI. Questa applicazione permette di costruire una sequenza digeribile da NI Veristand in maniera rapida e semplice, senza che sia richiesta una particolare conoscenza di programmazione.

Il processo di creazione consiste nel dichiarare le variabili che verranno utilizzate dalla sequenza, allacciando ciascuna di esse ad un canale presente nel System Definition File del progetto VS. Dunque si procede a popolare un diagramma di flusso con una serie di istruzioni, che possono esser suddivise in:

- **Espressioni:** in qualsiasi punto del codice è possibile definire il valore di una delle variabili dichiarate o implementare con esse delle formule matematiche.
- **Strutture:** ogni sequenza può esser arricchita con cicli for o while, piuttosto che con istruzioni condizionali (if-else) o strutture multitasking, in cui più operazioni vengono eseguite in parallelo.

- Sequenze: all'interno di una sequenza principale è possibile richiamare l'esecuzione di altre sequenze configurate in precedenza.

Una volta completata la scrittura del ciclo automatico, l'esecuzione di quest'ultimo può essere comandata o attraverso l'editor stesso, oppure da un tasto, opportunamente configurato per richiamare il test, sull'interfaccia di VS. La sequenza viene presa in carico dal NI Veristand Engine che si occupa di eseguirla in Real-Time alla frequenza di esecuzione del Primary control loop. La Figura 6-1 mostra la schermata dello Stimulus Profile editor durante la costruzione di una sequenza per la calibrazione anticipo. L'interfaccia può essere suddivisa in tre sezioni:

- sulla destra è presente la colonna in cui vengono dichiarate le variabili e i parametri che si vogliono utilizzare nella scrittura della sequenza. Selezionando le singole voci, il "property browser" che si apre a destra permette di configurare alcune impostazioni tra cui il canale del progetto VS a cui il parametro della sequenza può essere allacciato;
- al centro dello schermo è presente il flow chart che può essere popolato con le istruzioni. Quando si comincia a scrivere una nuova sequenza sul diagramma di flusso appaiono già tre sezioni in cui è possibile strutturare il codice: il *Setup* è dove si inizializzano i parametri, mentre all'interno del *Main* si implementa il corpo principale del codice. Infine il *CleanUp* può essere utilizzato per inserire delle istruzioni da eseguire come recovery, infatti Veristand prevede due modalità di interazione con le Real-Time sequences nel caso in cui scatti un allarme: *abort*, che termina immediatamente la sequenza, e *stop*, che, a prescindere dal punto a cui è arrivata l'esecuzione del ciclo, salta tutto il resto di codice fatta eccezione per la sezione del *CleanUp* che può appunto esser utilizzata per gestire un eventuale situazione di pericolo riportando ad esempio il motore in condizioni di sicurezza prima di uscire dall'automazione;
- gli elementi con cui si può costruire il diagramma di flusso (variabili, espressioni e sequenze precompilate) sono nella colonna di sinistra, suddivisi in librerie.

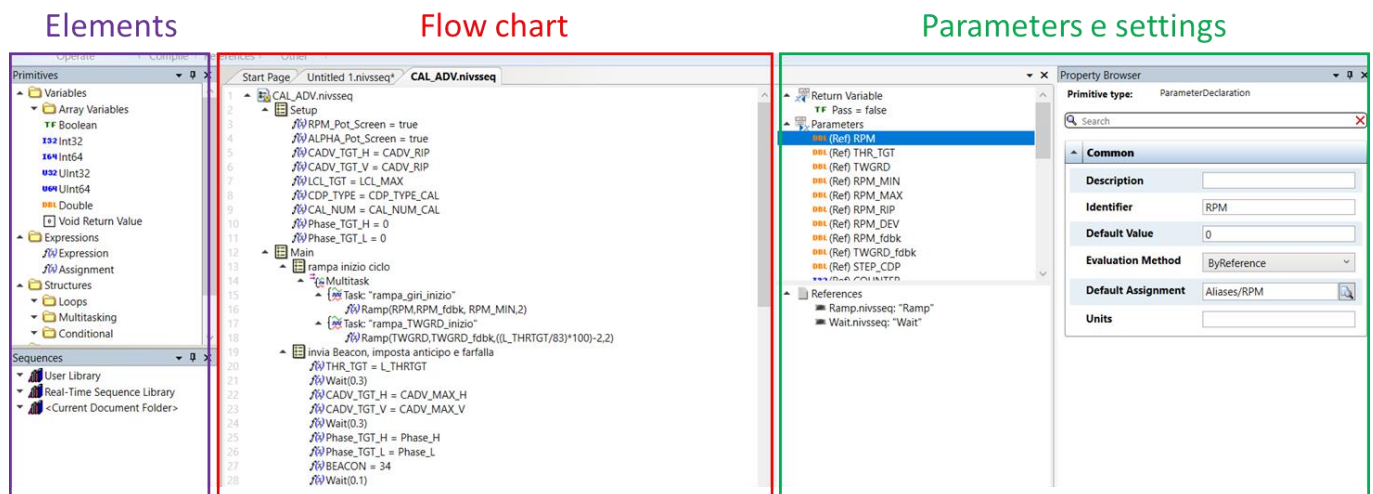


Figura 6-1: Una sequenza configurata su NI Stimulus Profile editor

Solitamente in sala prove non è richiesta l'implementazione di cicli particolarmente complessi in termini di profilo dei parametri da sovrascrivere. Anzi, di solito le varie attività di calibrazione e sviluppo, piuttosto che la rilevazione delle prestazioni dei motori, possono riassumersi come delle sequenze di rampe e stazionari di carico e giri motore. Dunque, lo Stimulus Profile editor risulta essere un ottimo strumento per sviluppare questo genere di automazioni, permettendo di creare in maniera semplice e veloce le sequenze RT grazie all'interfaccia a diagramma di flusso.

I limiti di tale metodo emergono durante la costruzione di sequenze con profili complessi, come ad esempio nel caso delle simulazioni del giro pista, in cui le condizioni di funzionamento da replicare al banco sono altamente variabili. Non essendo possibile sviluppare applicazioni di questo tipo assemblando un diagramma di flusso, per queste automazioni si ricorre ad una API (Application Programmable Interface) messa a disposizione da NI, che permette di costruire le sequenze per VS dall'ambiente di sviluppo di Labview. Tra le funzionalità di questa API c'è anche la possibilità di scrivere un ciclo a partire da un file di tipo CSV (comma separated values). Gli unici requisiti da rispettare consistono nell'indicare nella prima riga i nomi dei canali da replicare con l'automazione e nella prima colonna "timestamp" gli istanti temporali con cui viene scandita la sequenza. I vari canali del CSV (cioè le colonne) vengono interpretati come vettori di elementi che la sequenza andrà a scorrere durante l'esecuzione, immagazzinando ad ogni passo temporale il rispettivo valore di ogni canale all'interno di un parametro. Ogni parametro, relativo ad un singolo canale di

telemetria, viene creato all'atto della compilazione (eseguita questa volta tramite Labview) della sequenza e viene poi collegato, tramite l'editor, al canale del progetto VS che si intende sovrascrivere durante l'automazione. Così facendo si ottiene un player dei dati telemetria in pochi e semplici passaggi. La sequenza creata dall'API può poi essere modificata tramite lo Stimulus Profile editor per inserire particolari condizioni di ingresso e uscita dal ciclo, oppure per inserire l'istruzione di ripetere il giro pista un numero di volte che può essere configurato dall'interfaccia utente.

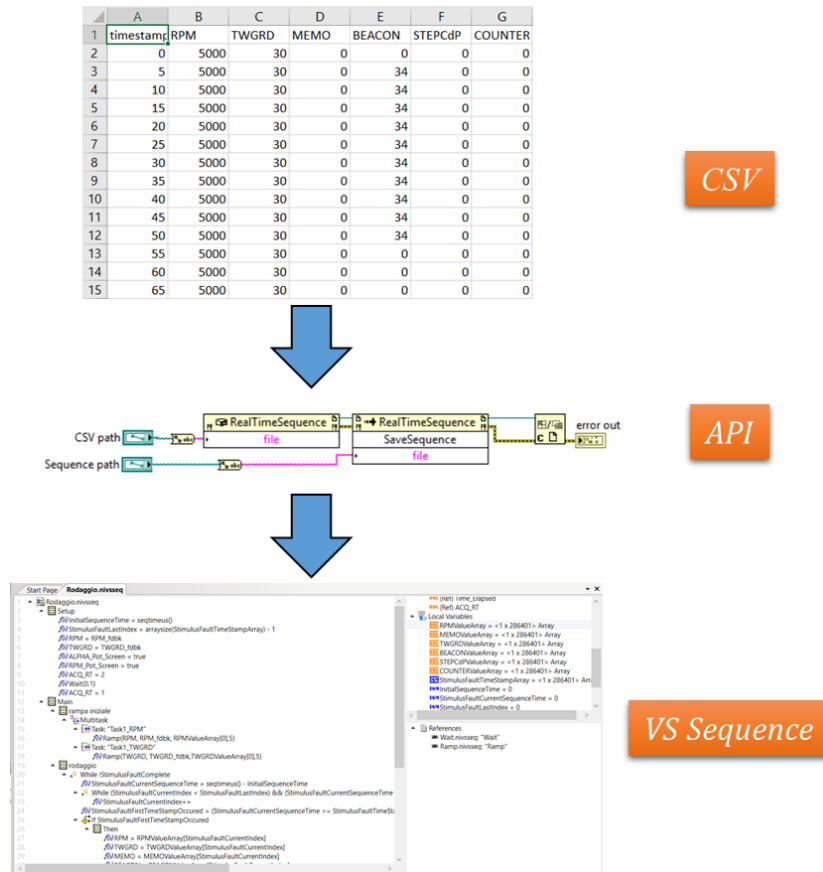


Figura 6-2: Costruzione della sequenza per la simulazione del giro pista

6.2 Test Manager tool

Lo svolgimento dei test di calibrazione e sviluppo motore ha reso necessario la creazione di un apposito tool di supporto all'operatore durante queste attività, le quali richiedono un livello di interattività con l'utente che NI Veristand non riesce ancora ad offrire. Questo genere di

prove consiste nell'esecuzione, ripetuta per un certo numero di volte, di semplici profili di carico e giri motore, sovrascrivendo, al tempo stesso, specifici parametri di controllo motore per mettere in evidenza l'effetto della variazione di questi ultimi sul comportamento del motore (prestazioni, emissioni, affidabilità, ecc..). Tra una ripetizione e l'altra, i profili dei parametri che vengono sovrascritti in centralina, così come quelli di carico e regime di rotazione, sono fissi e non cambiano, tuttavia i valori possono essere diversi ad ogni step. Dunque, benchè la forma dell'automazione rimanga immutata e rigida, l'operatore deve poter configurare, una volta per tutte e prima di avviare il ciclo automatico, tutti i valori dei parametri per ogni passo della sequenza. Data la possibilità di avere anche un numero considerevole di passi da configurare per una singola automazione, le configurazioni dei cicli devono poter essere salvate in modo da poter essere ricaricate eventualmente in futuro, velocizzando i tempi di configurazione ed evitando eventuali errori durante questa fase. Inoltre, al termine di ogni passaggio, ci sono alcune automazioni che devono attendere il consenso dell'operatore prima di passare allo step successivo, per ricevere informazioni utili per proseguire o perché non è scontato che si debba procedere al prossimo passo. Invece, altre automazioni devono procedere in totale autonomia, salvo rilevazione di eventuali errori, assicurando rapidità di esecuzione e maggiore ripetibilità della prova anche durante le pause tra due passi consecutivi (ad esempio per assicurare un tempo fisso di riposo per il raffreddamento degli scarichi dopo aver eseguito combustioni ritardate).

Per permettere all'utente di poter gestire le sequenze Real-Time con un tale livello di interazione, senza perdere mai di vista la flessibilità e la possibilità di adattamento a una sequenza configurabile qualsiasi, è stato necessario implementare un'applicazione in grado di fare da ponte di collegamento tra NI Veristand e le automazioni create con lo Stimulus Profile editor, ovvero il Test Manager (TMan). Questo tool offre all'utente un'interfaccia estremamente semplificata tramite cui è possibile selezionare da una raccolta di sequenze precompilate quella che si vuole eseguire, configurare ogni iterazione del ciclo e lanciarlo, monitorando poi l'esecuzione un passo alla volta.

6.2.1 Struttura

Il Test Manager è un applicativo, sviluppato in Labview, che viene eseguito sul pc Host: riesce ad interagire con il progetto Veristand in esecuzione sul target Real-Time grazie a delle API che sfruttano l'interfaccia utente di VS per vari scopi, come appunto comandare l'esecuzione di sequenze Real-Time, monitorare canali del progetto o fare acquisizione dati ad elevate frequenze (data streaming), modificare il System Definition File e molto altro ancora utilizzando codice implementato nel linguaggio grafico di Labview.

La struttura utilizzata per la scrittura del codice è stata scelta sulla base dei requisiti fondamentali dell'applicazione: semplicità di utilizzo per l'utente, flessibilità, robustezza e affidabilità. Dovendo proporre all'operatore un oggetto estremamente facile da utilizzare, si è scelto di implementare l'applicazione come un flusso rigido di schermate, strutturando il codice con una macchina a stati. Per non penalizzare la flessibilità e permettere all'eseguibile di adattarsi alle diverse automazioni si è cercato di strutturare le schermate in maniera molto generale, creando oggetti ad interfaccia che assicurassero il grado richiesto di personalizzazione delle automazioni. Dovendo implementare un'applicazione che eseguisse dei comandi sulla base delle istruzioni fornite dall'operatore ma che allo stesso tempo fosse in grado di prendere decisioni per agire poi in autonomia, non è stata utilizzata una macchina a stati semplice, bensì si è scelto di implementare due loop:

- un producer loop solitamente eseguito a bassa priorità, il cui ruolo è quello di ricevere le richieste fatte dall'operatore attraverso l'interfaccia e sulla base di queste generare dei comandi da eseguire. L'operatore non è il solo che può richiedere l'esecuzione di azioni, infatti anche il software stesso può farlo, sulla base dei dati a sua disposizione. In quest'ultimo caso la richiesta di un comando può essere partorita anche in parti del codice differenti dal producer loop.
- un consumer loop che riceve le richieste generate e le esegue il più velocemente possibile. Alcune delle richieste evase dal consumer possono essere state generate in questo stesso loop (decisioni prese dal software in autonomia).

Le azioni da intraprendere vengono sequenzialmente inserite in una coda, a prescindere dal fatto che vengano richieste dall'operatore o dal codice stesso. In questo modo i comandi

vengono eseguiti con un ordine ben preciso e si scongiura il pericolo che una richiesta venga persa senza essere evasa.

La Figura 6-3 mostra la struttura di una macchina stati Queue driven con producer e consumer loop nel linguaggio grafico di Labview.

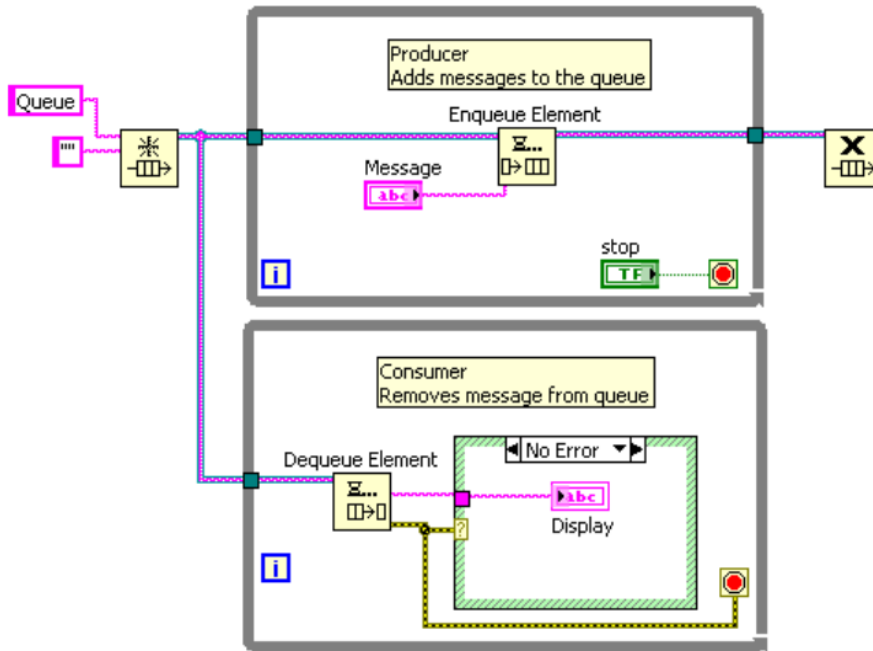


Figura 6-3: Struttura della macchina a stati queue driven con producer e consumer loop

6.2.2 Interfaccia

In questo paragrafo viene presentata l'interfaccia dello strumento sviluppato per la configurazione e l'esecuzione dei cicli automatici. La sequenza di schermate che guidano l'operatore, prima durante la fase di impostazione dei parametri e poi nel lancio dell'esecuzione del test, è la stessa per tutti i cicli. Alcune schermate possono tuttavia contenere degli oggetti che sono specifici, e quindi visualizzabili, solo per una determinata automazione. In questo paragrafo verrà esplorata l'interfaccia del Test Manager, simulando la configurazione di un ciclo per la calibrazione dell'anticipo.

Home: nella schermata principale, la prima che viene visualizzata all'apertura dell'applicazione, sono presenti le varie automazioni che sono state costruite attraverso lo

Stimulus Profile Editor. In alto a sinistra, sono presenti alcuni campi che possono essere riempiti dall'operatore con delle informazioni che verranno riportate ad esempio negli eventuali file di acquisizione. In basso a destra, il tasto per chiudere l'applicazione, accessibile, per ragioni di sicurezza, solo da questa schermata. Sempre dalla *Home* si può accedere, tramite la barra di menù, alla pagina dei *General Settings*: da qui è possibile impostare i percorsi delle sequenze da eseguire e la cartella dove salvare gli eventuali file di acquisizione.

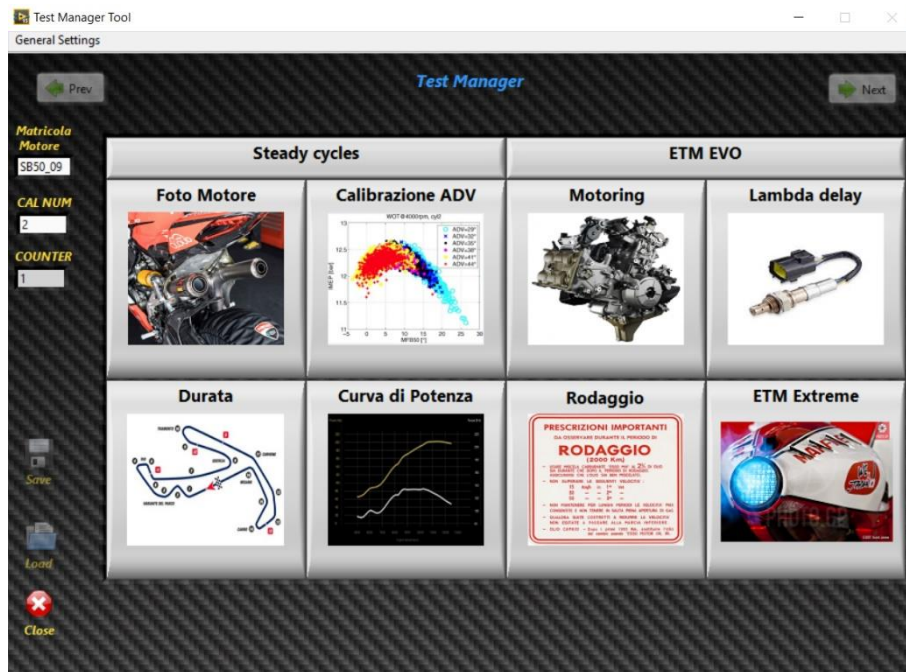


Figura 6-4: TMan – Home

Cycle Settings: una volta scelta l'automazione da eseguire si passa alla schermata delle impostazioni generali, ovvero valide per tutto il ciclo, non specifiche del singolo passo. Solitamente qui si configurano i valori da applicare ai parametri durante la fase di riposo che sta tra due passi consecutivi: manopola e regime di rotazione, correzione anticipo, farfalla, correzione lambda, ecc. Viene mostrata una preview del ciclo con alcune delle grandezze più importanti: manopola, apertura farfalle e giri motore. In questa schermata si attivano anche i due pulsanti, ai lati in alto, tramite i quali si può scorrere tra le schermate, mentre in basso a sinistra vengono abilitati i tasti per salvare e configurare la configurazione del ciclo.

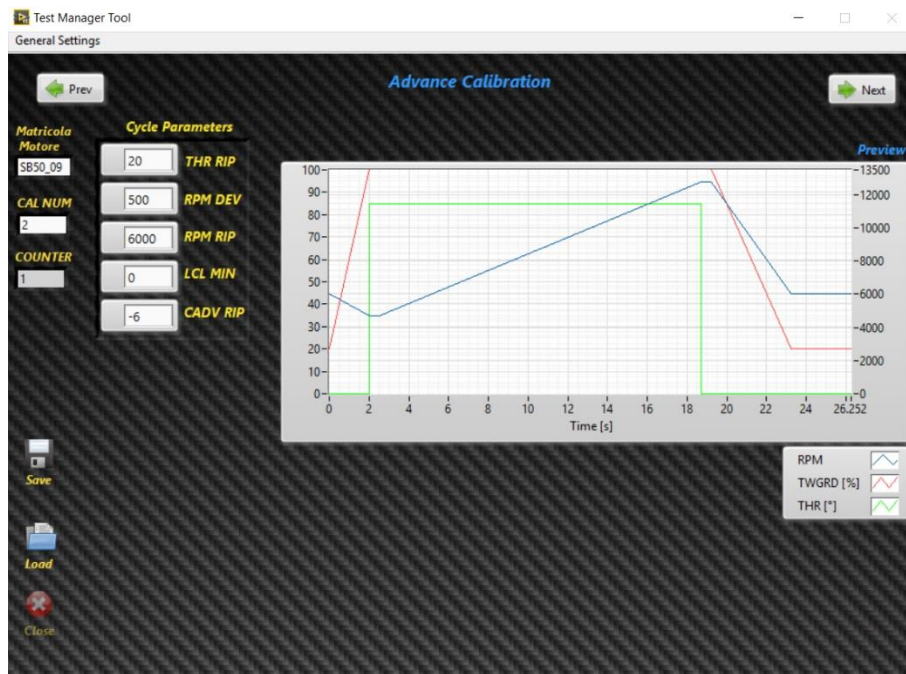


Figura 6-5: TMan: Cycle Settings

Step Settings: in questa schermata è possibile configurare i valori dei parametri utilizzati dalla sequenza per ogni passo dell'automazione. La prima colonna a sinistra, nel riquadro centrale, mostra tutti i parametri a cui possono avere accesso tutti i cicli presenti nella raccolta visibile nella schermata di *Home*.

Per ogni passo è possibile definire il regime minimo e quello massimo. Solitamente il profilo del regime motore per tutte le automazioni di calibrazione e sviluppo motore assume una forma simile ad un trapezio: a partire da un regime minimo si sale con una rampa avente una certa pendenza (definita in *Cycle Settings*), poi, dopo aver stazionato per un certo lasso di tempo a regime massimo si riporta il motore a regime minimo e analogamente negli step successivi. Tra un passo e quello seguente il motore viene sempre portato in condizioni di riposo: condizioni di regime e manopola o farfalla definite in *Cycle Settings*. Il regime di riposo può essere diverso da quello minimo definito per ogni step.

Fatta eccezione per il regime motore massimo e minimo e per l'ultima riga in cui si indica il numero di ripetizioni da eseguire per il singolo step configurato, i restanti parametri sono quelli che possono essere sovrascritti in centralina. I parametri, come mostra la Figura 6-6: TMan: Step Settings, possono avere diversi colori:

- Rosso spento: sono quei parametri a cui la sequenza scelta non ha accesso, dunque non possono ovviamente essere configurati, infatti, per ogni passo, le righe relative a queste variabili appaiono come ingrigite.
- Rosso acceso: sono quei parametri a cui la sequenza ha accesso ma che l'utente non vuole che siano sovrascritti durante il ciclo, dunque, anche in questo caso, le righe relative a queste variabili rimangono ingrigite. In questo caso la sequenza scrive a centralina dei valori che non comportano una sovrascrittura di queste variabili (correzioni nulle o valori che sono fuori da un range di accettabilità per la ECU)
- Verde: sono quei parametri a cui la sequenza ha accesso e l'utente vuole effettivamente sovrascrivere durante l'esecuzione dell'automazione. Un parametro che appare rosso acceso può essere abilitato alla sovrascrittura dall'utente: il parametro diviene verde e viene resa accessibile la relativa riga, permettendo la configurazione dei valori da sovrascrivere ad ogni step.

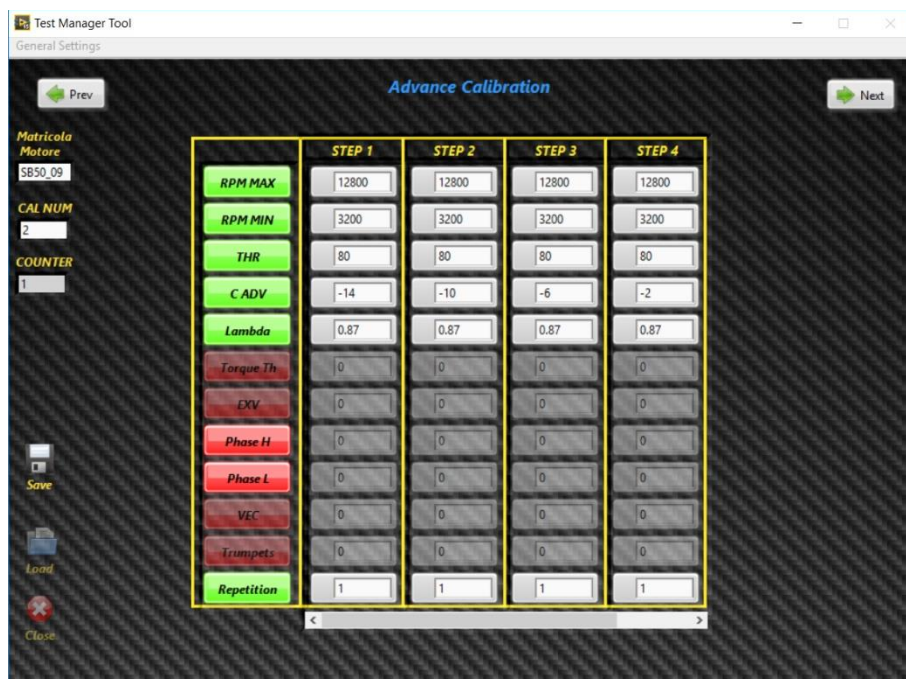


Figura 6-6: TMan: Step Settings

Tx Check: terminata la fase di configurazione dei vari parametri, prima di avviare il ciclo è necessario verificare che sia attiva la comunicazione con la ECU e che quest'ultima sia pronta

a ricevere via CAN i valori per sovrascrivere le variabili desiderate (funzionalità che si attiva attraverso un flag inviato dal software di gestione della centralina, un programma sviluppato da Magneti Marelli denominato Sysma, alla stessa ECU). Dunque, banalmente, quello che si fa è provare a scrivere a ECU un valore di lambda target molto diverso dal comune: se entro un secondo si rilegge da centralina il valore di lambda effettivamente impostato al valore sovrascritto allora si può procedere, al contrario non è concesso di andare oltre questa schermata (problemi di connessione CAN o mancata abilitazione alla sovrascrittura). Nel passaggio da questa schermata alla successiva il motore viene portato alle condizioni di riposo indicate all'inizio della fase di configurazione: l'ultimo passo prima di lanciare l'esecuzione del primo passo dell'automazione.



Figura 6-7: TMan - Tx Check

Check before Step: prima che venga eseguito ogni passo dell'automazione viene mostrata questa schermata per riepilogare le temperature misurate agli scarichi e presentare le attuali condizioni dell'aria comburente in termini di umidità e temperatura, confrontandole con le condizioni desiderate. Se tra un passaggio e il successivo questa schermata mostra dei parametri dell'aria troppo lontani dai valori target o delle temperature allo scarico troppo elevate, l'operatore può permanere in questa schermata, attendendo che il sistema di condizionamento dell'aria comburente faccia il suo dovere, riportando i valori entro un certo

range di validità rispetto ai desiderati oppure che gli scarichi semplicemente si raffreddino (ad esempio nel caso in cui siano stati eseguiti dei passi in cui la combustione era ritardata), dopodichè può dare inizio all'esecuzione del passo quando più lo reputa opportuno. Nel caso in cui l'automazione preveda l'esecuzione dei vari step in totale autonomia, questa schermata viene comunque mostrata, tra un passaggio e il successivo, per un breve lasso di tempo entro il quale l'operatore, toccando un punto qualsiasi dello schermo, può interrompere il lancio automatico dello step seguente. Al contrario, se l'operatore non blocca il proseguimento della sequenza e le condizioni dell'aria e degli scarichi sono ritenute accettabili per lo svolgimento della prova, il TMan prosegue in totale autonomia al lancio del passo successivo e così via, fermandosi solo al termine della sequenza. Da questa schermata è anche possibile abortire l'intera sequenza e ritornare alla schermata di *Home* se l'operatore lo ritiene necessario.

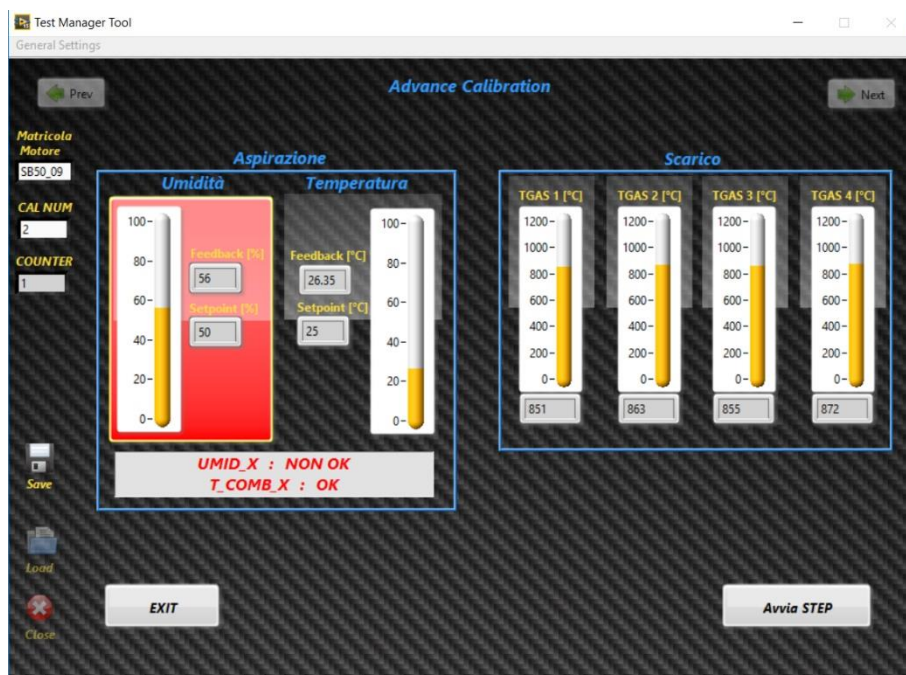


Figura 6-8: TMan - Check before step

Step Execution: Infine, una volta lanciata l'esecuzione del singolo passo, viene mostrata questa schermata che riepiloga i parametri che vengono sovrascritti durante il corso dello step e il profilo di giri motore, manopola e farfalle. Solitamente questa schermata viene personalizzata con dei contenuti specifici dell'automazione eseguita. Una volta terminata l'esecuzione del passo, nel caso in cui sia richiesta un'interazione con l'operatore (come nel caso della calibrazione anticipo), è l'utente che decide quando passare allo step successivo

premendo il tasto in basso a destra, in alternativa, l'automazione va avanti in autonomia. In entrambi i casi, ci si riporta prima alla schermata *Check before Step* e poi viene lanciato il passo successivo (dall'operatore, dove previsto, oppure direttamente dal TMan). Così si prosegue fino al termine della sequenza, ricordata all'operatore da un pop-up, dopodichè si ritorna alla schermata iniziale.

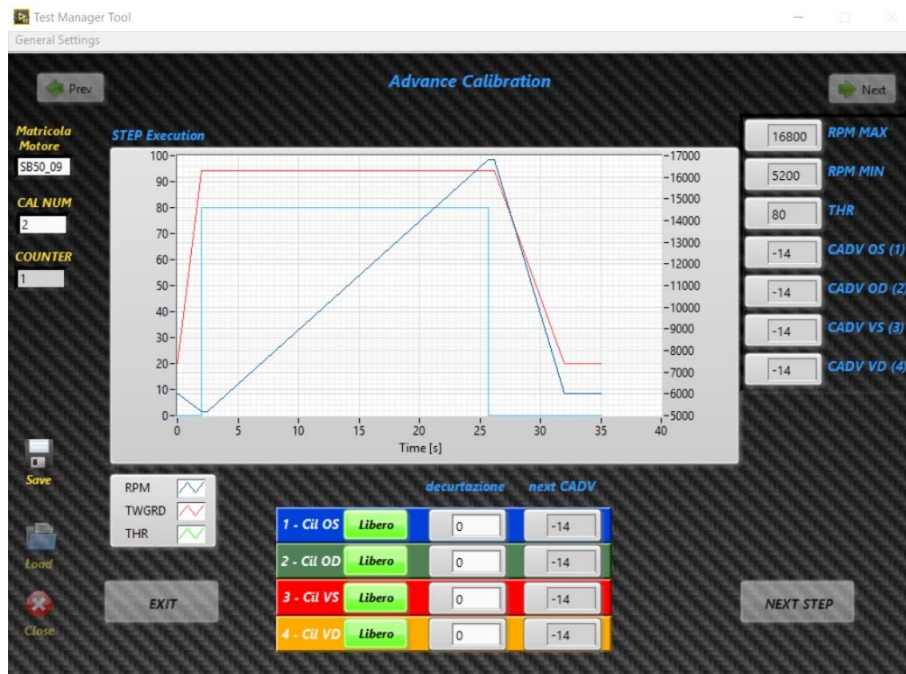


Figura 6-9: TMan - Step execution

6.3 La calibrazione anticipo

In questo paragrafo viene brevemente descritta la procedura di calibrazione dell'anticipo di accensione, come esempio di automazione che è possibile configurare ed eseguire con l'ausilio del tool descritto nella sezione precedente. Tra quelle realizzate, si è scelto di descrivere quest'automazione poiché durante il corso dell'attività di dottorato è stata oggetto di un grande sviluppo, volto a rendere ancora meno manuale la procedura, accorciando i tempi di esecuzione e diminuendo anche i rischi di detonazione eccessiva.

6.3.1 Calibrazione classica

La calibrazione dell'anticipo di accensione prevede l'esecuzione in un certo numero di passi, ciascuno dei quali consiste in una rampa di giri motore a carico costante (acceleratore costante, o nel caso si riescano a gestire direttamente le aperture della farfalle via CAN, farfalla costante). Vengono eseguite delle rampe in modo tale da poter analizzare l'intero range di funzionamento del motore per un determinato carico. Durante l'esecuzione dell'intero passo si apporta una correzione dell'anticipo di accensione rispetto al valore che verrebbe applicato dal controllo motore (derivante da mappe in funzione di regime e apertura farfalla). Ad ogni passo si applica una correzione anticipo diversa: più nello specifico si parte da una correzione negativa che provoca una combustione molto ritardata, per esser sicuri di non incappare in detonazione fin da subito, dunque si passa a ricentrare la combustione portandosi verso correzioni nulle e infine si correggono in positivo i valori di mappa, andando verso anticipi sempre più alti, fino a che non si termina la sequenza oppure si notano delle detonazioni particolarmente intense. La Figura 6-10 mostra l'esecuzione della procedura di calibrazione classica, evidenziando per ogni passo della sequenza la sola rampa crescente di giri motore. Dall'alto verso il basso i grafici mostrano il regime di rotazione, gli indici statistici di detonazione per i due cilindri, ottenuti applicando le correzioni di anticipo d'accensione mostrate nei due grafici in basso. Come si nota in figura, l'accrescimento della correzione applicata all'anticipo di mappa nel corso dei vari passaggi avviene in due fasi:

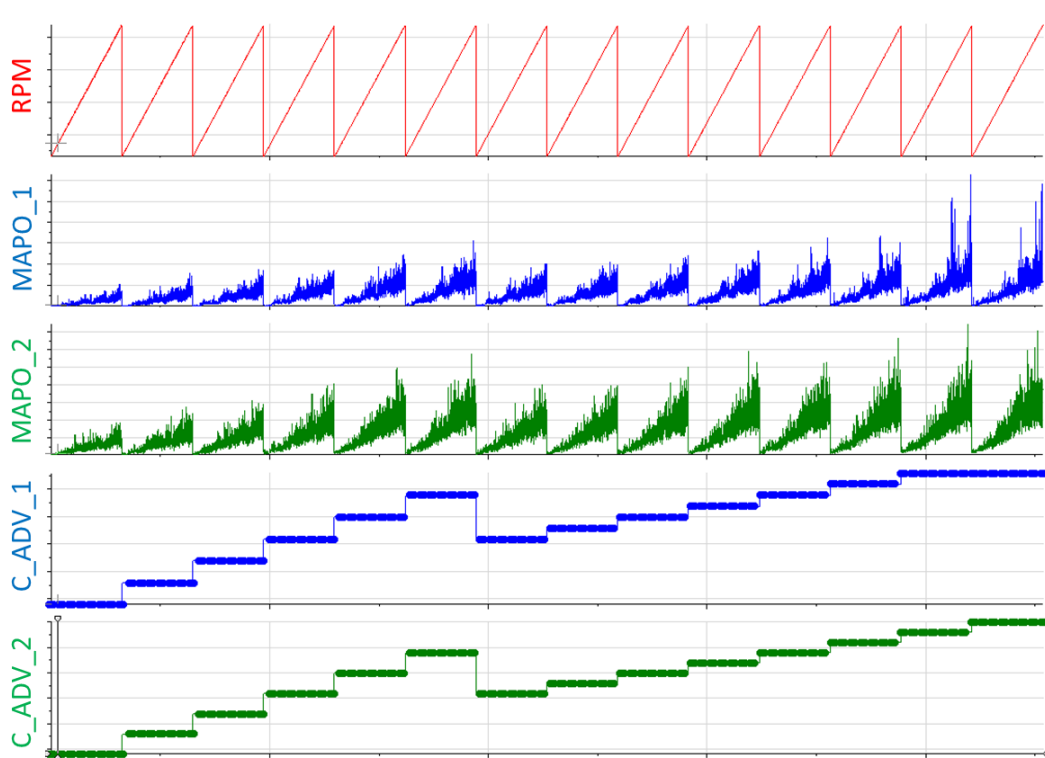


Figura 6-10: Calibrazione anticipo classica

- durante la prima fase si parte da una correzione molto negativa per arrivare al termine di questa fase ad aggiungere qualche grado all'anticipo di mappa, con un offset di correzione tra un passo e il successivo relativamente grande (4 gradi nel caso mostrato in figura): questa prima fase consente di indagare in maniera rapida come le prestazioni del motore si degradano applicando anticipi minori rispetto all'ottimale, un'analisi utile per imbastire logiche che ritardano appositamente la combustione per vari scopi (creare riserve di coppia o, in un caso diverso dall'ambito del motorsport, attivazione strategie di post-trattamento dei gas di scarico).
- Terminata la prima fase, durante la seconda ci si riporta ad una correzione leggermente negativa e, con offset di correzione minori rispetto alla prima fase (si aumenta di due gradi alla volta), si arriva a richiedere correzioni relativamente elevate (anche +12 gradi rispetto al valore di mappa) se l'intensità di detonazione non diviene troppo elevata, al contrario ci si ferma prima per evitare di danneggiare il motore. Come evidente in Figura 6-10 dai grafici del MAPO, le ultime rampe di questa fase

sono le più critiche per il motore, a causa delle elevate correzioni anticipo che vengono richieste alla ECU.

La configurazione dei parametri per una calibrazione anticipo classica viene mostrata dalla Figura 6-6: TMan: Step Settings. Figura 6-6: ad ogni step viene eseguita una rampa di giri da 3200 a 12800 rpm con apertura farfalla sempre a 80° e lambda target sovrascritto a 0.87. Si comincia nel primo passo ad applicare una correzione anticipo pari a -14 per poi andare via via ad aumentare (inizio della prima fase sopra descritta).

Come mostra la Figura 6-9, al termine di ogni passo del ciclo di calibrazione, viene mostrato un piccolo riquadro, al centro in basso, in cui è possibile specificare, per ciascun cilindro, su quale di questi si desidera bloccare la correzione anticipo al valore attuale per i prossimi passi. Infatti, questa tipologia di calibrazione, comporta ancora un elevato grado di manualità: l'operatore deve osservare in tempo reale i risultati del sistema di analisi combustione durante la rampa di giri motore e nel caso in cui si accorga che su un cilindro la correzione anticipo attuata dall'automazione comporti delle combustioni detonanti, può bloccare su quel cilindro l'aumento della correzione anticipo che verrebbe attuata nei passaggi successivi. Non solo si può bloccare la correzione anticipo al valore del passaggio appena effettuato, ma si può anche decidere di quanto decurtare tale valore per i prossimi passi. Infatti, se si blocca la correzione è perché ci si è già accorti di un livello di detonazione troppo elevato, dunque si vuole evitare di replicare un'intensità di detonazione simile a quella appena riscontrata.

6.3.2 Calibrazione EVO

Il metodo di svolgimento della calibrazione anticipo presentato nel paragrafo precedente presenta alcuni limiti:

- La procedura richiede grande impegno da parte dell'operatore. Infatti, come precedentemente descritto, la decisione di non proseguire ad incrementare la correzione anticipo su un cilindro nei passi successivi, per evitare di danneggiare il motore spetta solo all'operatore, che deve essere abile nel capire dai dati pubblicati dal sistema di analisi di combustione durante il corso della rampa, se si è raggiunto o meno un livello di detonazione eccessivo. Il metodo richiede troppo apporto manuale:

con motori che dispongono di un numero di cilindri superiore a due, l'analisi dei dati in tempo reale diviene sempre più difficile e il rischio è quello di non accorgersi visivamente di quanto ciascun cilindro sta detonando.

- La sovrascrittura via CAN permette di accedere solo a quei parametri che sono stati previsti durante la scrittura del software di controllo della centralina: solitamente si tratta di pochi parametri, che all'interno del software ECU costituiscono degli offset o dei guadagni di alcuni di quei parametri strategici che vengono applicati dal controllo. Il fatto di non aver accesso a tutte le variabili del software di controllo motore è molto limitante per svolgere una calibrazione completa e dettagliata. Inoltre, come avviene ad esempio nel caso dell'anticipo di accensione, potendo controllare solo il valore di una correzione rispetto all'attuazione base e non i singoli valori della mappa di gestione dell'anticipo, il risultato è che si applica un offset a tutta la mappa. Durante una rampa di calibrazione, in cui si analizza tutto il campo di funzionamento del motore, potrebbe capitare che ad un certo regime sia necessario correggere il valore di mappa aumentandolo e viceversa ad un altro regime. Tuttavia, potendo applicare una correzione unica a tutta la mappa, capiterà inevitabilmente a pari correzione applicata di produrre combustioni ritardate ad un certo regime (temperature elevate agli scarichi) e magari sovranicipate (pericolo di detonazione) ad un altro regime.

Per riuscire ad andare oltre i limiti appena descritti è stato necessario implementare un sostanziale upgrade del dialogo tra la centralina e il Test Manager. Grazie a delle API che sfruttano il protocollo “.net” è stato possibile interfacciare l'applicativo del TMan con il software di gestione della centralina (Sysma). Così facendo si è reso possibile l'accesso a tutte le mappe e alle singole variabili presenti nel software di controllo motore dal Test Manager, sia in lettura che in scrittura. Potendo accedere direttamente alla mappa per la gestione dell'anticipo di accensione, la procedura di calibrazione è stata profondamente rivista. Inizialmente viene eseguito un certo numero di rampe applicando ad ogni passo una diversa correzione, costante per tutti i valori di mappa di anticipo da calibrare (cioè quelli relativi alla condizione di carico da analizzare): questa fase è necessaria per ottenere una certa statistica, infatti, al termine di ogni rampa viene eseguito un algoritmo che, in pochi secondi, processa i dati indicating (PMI, MFB50, indici statistici di detonazione...) acquisiti durante il passo

appena eseguito, stabilendo per ciascun cilindro e per ogni breakpoint della mappa di gestione dell'anticipo le nuove calibrazioni ottimali. Al termine delle prime rampe necessarie alla creazione della statistica, i risultati prodotti dall'algoritmo vengono utilizzati per sovrascrivere i vecchi valori presenti in mappa. Dunque si prosegue iterativamente ad eseguire rampe, calcolare le nuove calibrazioni scaricando i risultati in centralina. L'obiettivo è quello di coprire per ogni cilindro il maggior campo di PMI-MFB50, raggiungendo una calibrazione ottimale dell'anticipo di accensione, evitando durante la procedura di mettere il motore sotto stress con combustioni troppo ritardate o troppo anticipate. Questo approccio di calibrazione consente di ottenere una corretta mappatura dell'anticipo, per una singola condizione di carico, già dopo l'esecuzione di 10-15 rampe di giri motore: i tempi di esecuzione dell'attività vanno dai 20 ai 30 minuti. Una calibrazione completa prevede l'esecuzione della sequenza descritta per diverse aperture delle farfalle: in questo modo è possibile compilare l'intera mappa di correzione anticipo. Un ulteriore vantaggio di questo approccio rispetto a quello classico consiste nel fatto che l'apporto dell'operatore di cella viene ovviamente ridotto, così come viene ridotto (praticamente annullato) il lavoro di post-trattamento dei dati per ottenere le nuove calibrazioni, dati i risultati ottenuti con l'algoritmo di ottimizzazione.

Il dialogo con Sysma attraverso il protocollo “.net” è stato implementato all'interno del codice del TMan senza stravolgere la struttura precedente. In particolare, il controllo sulla scrittura verso centralina attraverso il Bus CAN viene sostituito dal controllo di connessione con Sysma. A tale scopo è necessario che il software di gestione della centralina sia installato direttamente sullo stesso pc dove viene eseguito anche il Test Manager che a sua volta è vincolato a stare sulla stessa macchina in cui è presente l'interfaccia utente del progetto Veristand. Dunque tutte e tre le applicazioni devono essere in esecuzione sul pc Host: solo così il TMan può comunicare con Sysma, per leggere i parametri di controllo motore e inviare nuove calibrazioni alla centralina. Volendo comunque mantenere Sysma anche su un pc separato (pc Host ECU) per agevolare certe operazioni, come lo scarico dati da ECU al termine della prova, senza ostacolare il controllo del banco sul pc Host, basta semplicemente avere entrambi i pc Host nella stessa rete IP della centralina, in modo tale che essa possa comunicare con entrambi.

Anche l'algoritmo di calibrazione dell'anticipo è stato facilmente introdotto all'interno della struttura del TMan integrando le relative schermate per l'interfacciamento con l'utente, come quella mostrata in Figura 6-11 che presenta il riepilogo delle rampe precedentemente eseguite: per ogni singolo breakpoint di regime che viene selezionato, vengono mostrati i punti sperimentali (e il relativo fitting) ottenuti in termini di PMI e MAPO (indice di detonazione) al variare dell'anticipo attuato durante le rampe. L'algoritmo di calibrazione, scritto e compilato in Matlab, consiste in una serie di eseguibili che vengono richiamati in differenti punti del codice del Test Manager.

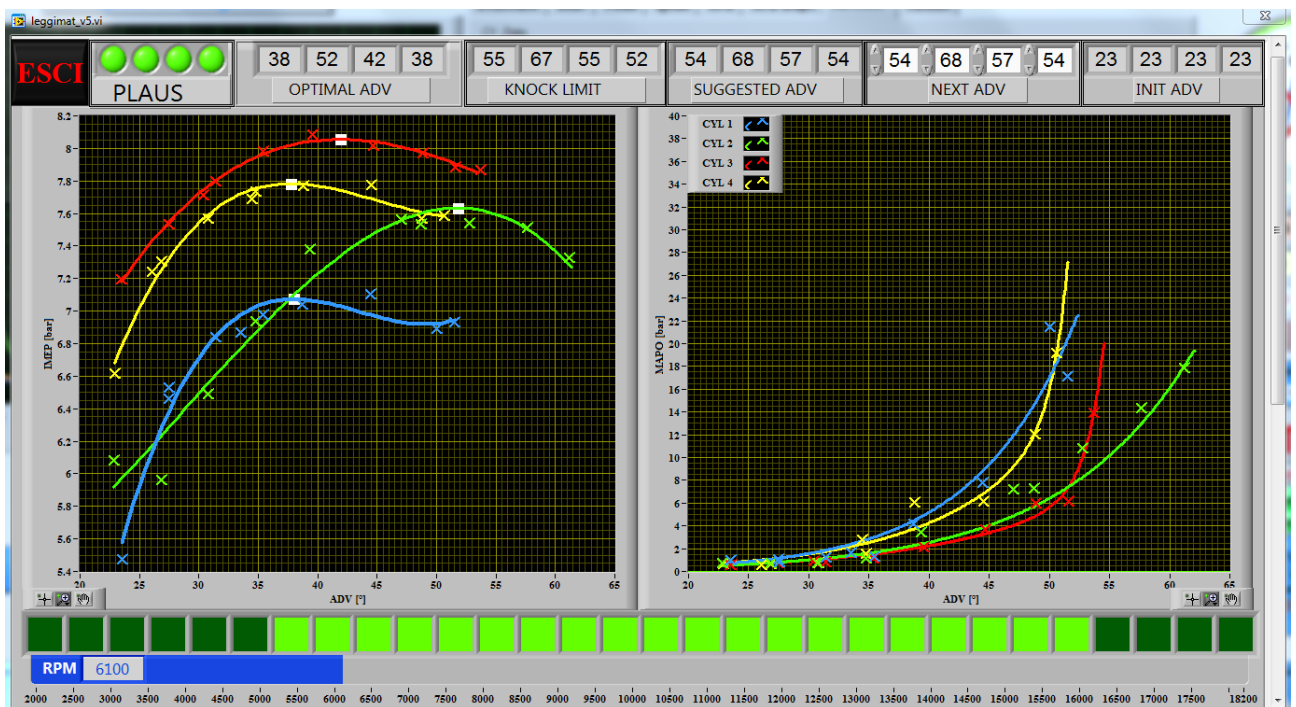


Figura 6-11: Interfaccia algoritmo di calibrazione

7. Conclusioni

Durante i tre anni di dottorato sono state affrontate diverse tematiche relative all'ambito della sperimentazione sui motori a combustione interna. Le varie attività condotte hanno portato alla creazione di un sistema per la gestione della sala prove flessibile, user-friendly, e a basso costo di sviluppo e gestione, che attualmente viene utilizzato per svolgere attività sperimentali nelle sale prove di Unibo e di Alma-Automotive. Il sistema è stato evoluto, integrando sempre più funzionalità, sviluppate per venire incontro alle esigenze di ogni banco prove in cui è stato utilizzato. Riassumendo, i temi affrontati durante lo sviluppo del sistema sono stati i seguenti:

- NI Veristand è stato scelto come ambiente di sviluppo per l'applicazione dedicata alla gestione della sala prove. Le funzionalità native sono state ampiamente esplorate e sfruttate per creare un'applicativo per il Real-Time testing sui motori a combustione interna. La natura flessibile e personalizzabile di NI Veristand ha permesso lo sviluppo di veri e propri add-ons, implementati in altri linguaggi di programmazione e inseriti all'interno dell'applicativo per integrare delle funzionalità aggiuntive richieste dalle esigenze specifiche di ogni sala prove gestita con il software sviluppato. Lo sviluppo degli add-ons per NI Veristand ha riguardato in parte l'implementazione di codici per lo svolgimento di calcoli in Real-Time e in parte la creazione di driver per il colloquio con i diversi devices presenti in sala prove.
- L'esigenza di controllare sistemi di condizionamento complessi, visto il numero di componenti su cui agire simultaneamente, ha portato allo sviluppo di una metodologia di gestione centralizzata di alcuni impianti della sala prove. Più in particolare, sono state implementate logiche per la gestione dei sistemi di condizionamento della temperatura dei fluidi motore (acqua e olio) e della temperatura e umidità dell'aria comburente. Data l'influenza che tali parametri possono esercitare sul comportamento del propulsore durante i test al banco, è stato necessario implementare dei modelli di controllo in grado di gestire con precisione le varie attuazioni degli impianti di sala al fine di mantenere le suddette variabili al valore desiderato, assicurando un elevato livello di accuratezza e ripetibilità. Questi obiettivi sono stati raggiunti implementando un approccio al controllo che unisce i benefici di un contributo model-based a quelli

dati da un controllo in retroazione esercitato da un classico PI. Il contributo in catena aperta, valutato sulla base di modelli fisici dei fenomeni da controllare, una volta calibrato riesce a supportare l'azione del PI, evitando di demandare a quest'ultimo l'onere della risposta dinamica, permettendo l'uso di guadagni contenuti. Una logica di controllo basata solo sulla retroazione comporterebbe una struttura del contributo in catena chiusa ben più complessa di un semplice PI, la quale richiederebbe inoltre un pesante lavoro di calibrazione. Le logiche di controllo sono state integrate nel sistema di gestione delle sale prove, ciò ha permesso di validare e calibrare gli algoritmi alimentandoli direttamente con i segnali del banco, svolgendo diversi tipi di test, facendo lavorare il motore sia in condizioni stazionarie che altamente transitorie. I risultati, in termini di raggiungimento e mantenimento dei valori target durante le varie prove, sono stati soddisfacenti. Si potrebbero ottenere dei miglioramenti sfruttando dei componenti hardware che possono essere gestiti in maniera più intelligente (richiedendo ad esempio un valore target della variabile controllata piuttosto che la semplice accensione/spegnimento del componente).

- Un modello del sistema di raffreddamento motore è stato sviluppato modellando i meccanismi di scambio termico tra motore, liquido di raffreddamento e aria che impatta sul radiatore. Un modello di questo tipo, oltre a risultare utile per dimensionare i componenti del sistema di raffreddamento, può essere d'aiuto anche durante la definizione delle strategie per il raffreddamento del motore. Tali logiche vanno ad agire sugli attuatori a disposizione nel circuito di raffreddamento installato sul veicolo per mantenere una certa temperatura target del liquido di raffreddamento. Anche il controllo dei sistemi di condizionamento dei fluidi motore in sala prove può beneficiare di un modello di questo tipo: gli algoritmi di gestione possono essere validati con delle semplici simulazioni, senza la necessità di utilizzare il motore reale.
- Infine, una parte rilevante dell'attività di dottorato è stata dedicata allo sviluppo di un tool per gestire diverse automazioni, utili all'attività di sperimentazione e calibrazione di motori da corsa. In particolare la calibrazione dell'anticipo di accensione è stata oggetto di uno studio approfondito che ha portato all'evoluzione dell'approccio di svolgimento di questa tipologia di calibrazione, accorciando notevolmente i tempi di

esecuzione di questa procedura e diminuendo al contempo i rischi di danneggiamento del propulsore a causa di eventuali combustioni sovranicipate o troppo ritardate. Ciò è stato possibile grazie al lavoro che è stato fatto per potenziare il colloquio tra il sistema di gestione della sala prove e la centralina di controllo motore.

In conclusione, l'esperienza maturata durante i tre anni nell'ambito della sala prove e dell'attività di sperimentazione in generale ha portato alla definizione di uno strumento che può essere sfruttato per svolgere attività di testing Real-Time general purpose grazie alle sue possibilità pressochè infinite di personalizzazione. La modellazione di componenti, lo sviluppo di logiche di controllo, automazioni e colloqui tra devices sono temi che oggi ricoprono un ruolo fondamentale in ambito automotive, dove la continua evoluzione dei propulsori implica un aggravio sempre maggiore della fase di sperimentazione.

Definizioni e abbreviazioni

T_{IA-TGT}	Target di temperatura aria in aspirazione motore.
$Urel_{TGT}$	Target di umidità relativa aria in aspirazione motore.
$Uabs_{TGT}$	Target di umidità assoluta aria in aspirazione motore.
T_{ENV}	Temperatura aria in aspirazione all'unità di condizionamento.
$Urel_{ENV}$	Umidità relativa aria in aspirazione all'unità di condizionamento.
$Uabs_{ENV}$	Umidità assoluta aria in aspirazione all'unità di condizionamento.
T_{IA}	Temperatura aria in aspirazione motore.
$Urel_{IA}$	Umidità relativa dell'aria in aspirazione motore.
P_{BARO}	Pressione barometrica.
\dot{m}_{VOL}	Portata in massa di vapore acqueo richiesta dal controllo umidità in catena aperta.
\dot{m}_{VCL}	Portata in massa di vapore acqueo richiesta dal controllo umidità in catena chiusa.
\dot{m}_V	Portata in massa di vapore acqueo richiesta in totale dal controllo umidità.
\dot{m}_{VSG}	Contributo di portata in massa di vapore acqueo prodotto dal generatore di vapore.
\dot{m}_{VCH}	Contributo di portata in massa di vapore acqueo prodotto dal chiller.
\dot{m}_{VCHmin}	Contributo di portata in massa di vapore acqueo minimo prodotto dal chiller.
\dot{m}_{VSGmin}	Contributo di portata in massa di vapore acqueo minimo prodotto dal generatore di vapore.
\dot{m}_{IA}	Portata in massa di aria in aspirazione al motore.
ΔT_{EC}	Variazione di temperature richiesta dal controllo di temperatura nella prima sezione dell'UTA.
ΔT_{CR2}	Variazione di temperature richiesta dal controllo di temperatura nella seconda sezione dell'UTA.
T_{PC}	Temperatura dell'aria misurata a valle della batteria di raffreddamento
\dot{H}_1	Potenza termica necessaria per ottenere nella prima sezione dell'UTA una variazione di temperatura pari a ΔT_{EC} .
\dot{H}_{2-OL}	Contributo in catena aperta della potenza termica richiesta alla resistenza di post-riscaldamento.
\dot{H}_{2-CL}	Contributo in catena chiusa della potenza termica richiesta alla resistenza di post-

	riscaldamento.
\dot{H}_2	Potenza termica totale richiesta alla resistenza di post-riscaldamento.
C_{IA}	Potenza termica specifico dell'aria.
CH_{cc}	Capacità di raffreddamento del chiller
SG	Input generatore di vapore: segnale analogico proporzionale alla portata di vapore da produrre.
SG_C	Capacità nominale del generatore di vapore.
$CH\%$	Duty cycle del segnale PWM inviato al chiller.
$WRpreH$	Potenza resistenza di pre-riscaldamento.
$RpostH\%$	Duty cycle del segnale PWM inviato alla resistenza di post-riscaldamento.
$RpreH\%$	Duty cycle del segnale PWM inviato alla resistenza di pre-riscaldamento.
f_{SG}	Fattore di scala per la conversione del segnale del generatore di vapore da unità ingegneristiche a elettriche.
$Uabs_{CH}$	Umidità assoluta desiderata a valle della batteria di raffreddamento.
\dot{H}_{O-OL}	Contributo in catena aperta del potenza termica da estrarre dall'olio motore.
\dot{H}_{E-O}	Potenza termica trasferita dal motore al lubrificante.
\dot{q}_{EO}	Portata olio motore.
C_{EO}	Potenza termica specifico olio motore.
ρ_{EO}	Densità olio motore.
T_{EO-TGT}	Target temperatura lubrificante all'uscita del motore.
T_{EO-OUT}	Temperatura lubrificante all'uscita del motore.
$T_{EO-C-OUT}$	Temperatura lubrificante all'uscita dello scambiatore olio motore/acqua di torre.
T_{DO-TGT}	Setpoint temperatura olio diatermico all'uscita del riscaldatore.
\dot{q}_{DO}	Portata olio diatermico.
c_{DO}	Calore specifico olio diatermico.
ρ_{DO}	Densità olio diatermico.
$T_{DO-H-OUT}$	Temperatura olio diatermico all'uscita dello scambiatore olio motore/olio diatermico.

ΔT_{O-CL}	Contributo in catena chiusa del controllo temperatura olio motore.
DEM_{O-pump}	Segnale di comando pompa acqua di torre del circuito di condizionamento dell'olio motore.
c_{CTW-O}	Calore specifico acqua di torre nel circuito di condizionamento dell'olio motore.
ρ_{CTW-O}	Densità acqua di torre nel circuito di condizionamento dell'olio motore.
$T_{CTW-OUT-O}$	Temperatura acqua di torre all'uscita dello scambiatore olio motore/acqua di torre.
T_{CTW-IN}	Temperatura acqua di torre all'uscita della torre evaporativa.
\dot{H}_{O-DIS}	Potenza termica dissipata dai tubi del circuito di condizionamento dell'olio motore.
\dot{H}_{O-TOT}	Potenza termica totale da rimuovere dall'olio motore.
\dot{H}_{O-CL}	Contributo in catena chiusa del potenza termica da estrarre dall'olio motore.
\dot{H}_{c-OL}	Contributo in catena aperta del potenza termica da estrarre dall'olio motore.
\dot{q}_c	Portata refrigerante.
C_c	Calore specifico refrigerante.
T_{c-TGT}	Target temperatura refrigerante all'uscita del motore.
T_{c-OUT}	Temperatura refrigerante all'uscita del motore.
T_{c-OUTr}	Temperatura refrigerante all'uscita del radiatore
T_{c-IN}	Temperatura refrigerante all'ingresso del motore.
T_{c-INr}	Temperatura refrigerante all'ingresso del radiatore.
ρ_c	Densità refrigerante.
\dot{H}_{E-c}	Potenza termica trasferita dal motore al refrigerante.
\dot{H}_{E-c}	Potenza termica ceduta dal motore al refrigerante.
\dot{H}_{c-E}	Potenza termica asportata dal motore per mezzo del refrigerante
\dot{H}_{c-TOT}	Potenza termica totale da rimuovere dal refrigerante.
\dot{H}_{c-CL}	Contributo in catena chiusa del potenza termica totale da rimuovere dal refrigerante.
\dot{H}_{c-OL}	Contributo in catena aperta del potenza termica totale da rimuovere dal refrigerante.

\dot{H}_{c-DIS}	Potenza termica dissipata dai tubi del circuito di condizionamento del refrigerante.
DEM_{c-pump}	Segnale di comando pompa acqua di torre del circuito di condizionamento dell'acqua motore.
c_{CTW-c}	Calore specifico acqua di torre nel circuito di condizionamento dell'acqua motore.
ρ_{CTW-c}	Densità acqua di torre nel circuito di condizionamento dell'acqua motore.
$T_{CTW-OUT-c}$	Temperatura acqua di torre all'uscita dello scambiatore acqua motore/acqua di torre.
P_o	Potenza erogata dal motore.
C_E	Capacità termica del motore.
\dot{m}_c	Portata in massa del refrigerante sul ramo pompa motore.
\dot{m}_r	Portata in massa del refrigerante sul ramo radiatore.
C_r	Capacità termica del radiatore.
\dot{H}_{c-r}	Potenza termica che il refrigerante trasferisce al radiatore.
\dot{H}_{r-a}	Potenza termica che il radiatore trasferisce all'aria.
\dot{H}_{r-aMAX}	Massima potenza termica che l'aria può teoricamente estrarre dal radiatore.
u	Coefficiente di scambio termico globale.
A	Superficie di scambio termico.
ε	Efficienza di scambio termico del radiatore.
\dot{C}_{max}	Capacità termica oraria maggiore.
\dot{C}_{min}	Capacità termica oraria minore.
NTU	Numero di unità di scambio termico.
T_{Ar}	Temperatura dell'aria che lambisce il radiatore.
K_{V1}	Resistenza idraulica della valvola che simula la valvola elettrostatica sul ramo di by-pass.
K_{V2}	Resistenza idraulica della valvola che simula la valvola elettrostatica sul ramo del radiatore.
K_r	Resistenza idraulica del radiatore.
K_E	Resistenza idraulica del motore.
K_{tot}	Resistenza idraulica totale sul circuito.

Δp_r	Caduta di pressione sul radiatore.
\dot{q}_r	Portata volumetrica di refrigerante sul ramo radiatore.
Δp_p	Incremento di pressione dato dalla pompa.
Q_p	Portata elaborata dalla pompa.
Δp_{bp}	Caduta di pressione sul ramo di by-pass.
Q_{bp}	Portata sul ramo di by-pass.
$\dot{m}_{air\ sp}$	Portata in massa di aria provocata dal movimento del veicolo.
sp_V	Velocità veicolo.
ρ_{air}	Densità aria che lambisce il radiatore
A_r	Sezione frontale radiatore.
ε_r	Efficienza aerodinamica radiatore.
TBS	Test Bench Supervisor
TBMS	Test Bench Measurement System
TBCS	Test Bench Conditioning System
TBAS	Test Bench Auxiliary System
ECU	Engine Control Unit
RT	Real-Time
FPGA	Field Programmable Gate Array
RTOS	Real-Time Operating System
VS	Veristand
NI	National Instruments
PCL	Primary Control Loop
SDF	System Definition File
SE	System Explorer

CD	Custom Device
I/O	Input/Output
CAN	Controller Area Network
FIFO	First IN First OUT
VI	Virtual Instrument
TMan	Test Manager
PMI	Pressione Media Indicata
PME	Pressione Media Effettiva

Riferimenti bibliografici

1. Michael Paulweber, Klaus Lebert, "Powertrain Instrumentation and Test Systems: Development – Hybridization – Electrification", Springer, 2016.
2. Martyr, A. and Plint, M., "Engine Testing – 4th Edition," Butterworth-Heinemann, March 2012.
3. Richard D Atkins, An Introduction to Engine Testing and Development, SAE International, 2009.
4. Friedrich, C., Compera, Y, Auer, M., Stiesch, G. et al., "An Efficient Test Methodology for Combustion Engine Testing: Methods for Increasing Measurement Quality and Validity at the Engine Test Bench", SAE Technical Paper 2017-01-0604, 2017, doi:10.4271/2017-01-0604.
5. National Instruments, «What is a Real-Time Operating System (RTOS)?» 22 11 2013. [Online]. Available: <http://www.ni.com/white-paper/3938/en/>.
6. National Instruments, «Configure Real-Time Testing Applications» 28 06 2017. [Online]. Available: <http://www.ni.com/white-paper/13068/en/>
7. National Instruments, «NI Veristand Engine Architecture» 16 08 2013. [Online]. Available: <http://www.ni.com/product-documentation/13033/en/>.
8. National Instruments, «Creating FPGA-Based I/O Personalities for NI Veristand» 29 12 2015. [Online]. Available: <http://www.ni.com/white-paper/9349/en/>.
9. Pizzonia, F., Castiglione, T., Bova, S., "A Robust Model Predictive Control for efficient thermal management of internal combustion engines", 2016, Applied Energy, Volume 169, pp. 555-566, DOI: 10.1016/j.apenergy.2016.02.063.
10. Castiglione, T., Pizzonia, F., Bova, S., "A novel cooling system control strategy for internal combustion engines", 2016, SAE International Journal of Materials and Manufacturing, Volume 9 Issue 2, pp. 294-302, DOI: 10.4271/2016-01-0226.

11. Bonkoski, P., Karnik, A., and Fuxman, A., "Calibration and Demonstration of Vehicle Powertrain Thermal Management Using Model Predictive Control," *SAE Int. J. Engines* 10(2):2017, doi:10.4271/2017-01-0130.
12. Karnik, A., Fuxman, A., Bonkoski, P., Jankovic, M. et al., "Vehicle Powertrain Thermal Management System Using Model Predictive Control," *SAE Int. J. Mater. Manf.* 9(3):2016, doi:10.4271/2016-01-0215.
13. Maciejowski J. M., *Predictive Control with Constraints*, Pearson Education Ltd. , 2002.
14. Taylor, O., Pearson, R., Stone, R., "Reduction of CO2 Emissions through Lubricant Thermal Management During the Warm Up of Passenger Car Engines", SAE Technical Paper 2016-01-0892, 2016, doi:10.4271/2016-01-0892.
15. E. Hovin, Edward C. Chiang, John H. Johnson, Design and Computer Simulation of Microprocessor Controlled Lubricating Oil Cooling System for Truck Diesel Engine; Technical Paper, 880488; ISSN 0148-7191; DOI: 10.4271/880488; Published February 1, 1988 by SAE International in United States
16. Shankar P. Bhattacharyya, Aniruddha Datta, and Lee H. Keel. (2009) "Linear Control Theory: Structure, Robustness, and Optimization", Automation and Control Engineering Series, CRC Press
17. Shankar P. Bhattacharyya, Ming-Tzu Ho, and Aniruddha Datta. (2000) "Structures and synthesis of PID controllers", *Advances in Industrial Control*, Springer
18. Kanefsky P., Nelson, V., Ranger M., "A Systems Engineering Approach to Engine Cooling Design", SAE Technical Paper, 1999-01-3780, 1999, doi: 10.4271/1999-01-3780.
19. Simmonds, N., Tsoutsanis, P., Drikakis, D., Gaylard, A. et al., "Full Vehicle Aero-Thermal Cooling Drag Sensitivity Analysis for Various Radiator Pressure Drops," SAE Technical Paper 2016-01-1578, 2016, doi:10.4271/2016-01-1578.

20. Shimizu, T., Abe, T., Sunayama, Y., Watanabe, S., Nakamura, E., "Simultaneous Evaluation on Aerodynamics and Air-cooling Performances for Motorcycle using CFD Analysis", SAE Technical Paper 2009-32-0138, 2009,
21. Jawad, B., Zellner, K., and Riedel, C., "Small Engine Cooling and the Electric Water Pump," SAE Technical Paper 2004-32-0084, 2004, <https://doi.org/10.4271/2004-32-0084>
22. Liu, G., Zhao, Z., Guan, H., Zhang, C. et al., "Benefit of Coolant Control Module for Thermal Management on SUV," SAE Technical Paper 2017-01-0152, 2017, <https://doi.org/10.4271/2017-01-0152>
23. Tao, X. and Wagner, J., "An Engine Thermal Management System Design for Military Ground Vehicle - Simultaneous Fan, Pump and Valve Control," SAE Int. J. Passeng. Cars – Electron. Electr. Syst. 9(1):243-254, 2016, <https://doi.org/10.4271/2016-01-0310>
24. Wagner, J., Ghone, M., Dawson, D., and Marotta, E., "Coolant Flow Control Strategies for Automotive Thermal Management Systems," SAE Technical Paper 2002-01-0713, 2002, <https://doi.org/10.4271/2002-01-0713>
25. Gulich, J., F., "Centrifugal Pumps", 2014, Springer, doi: 10.1007/978-3-642-40114-5
26. Cannon, R., H., "Dynamics of Physical Systems", 2003, Dover Publications, INC, Mineola, New York
27. Theodore L. Bergman, Adrienne S. Lavine, Frank P. Incropera, David P. DeWitt, "Fundamentals of Heat and Mass Transfer", John Wiley and sons, (2011), ISBN: 978-0-470-50197-9
28. L. Solieri, *Sviluppo di algoritmi avanzati di analisi e diagnosi combustione in tempo reale per motori endotermici alternativi*, PhD Thesis.
29. L. Zoffoli, *Sviluppo di Modelli Zero-Dimensionali per la Simulazione di Componenti di Motori e Veicoli su Piattaforme HiL*, PhD Thesis, 2018.