

SCUOLA DI INGEGNERIA E ARCHITETTURA

DIN

Dipartimento di Ingegneria Industriale

CORSO DI LAUREA MAGISTRALE IN INGEGNERIA MECCANICA

TESI DI LAUREA

In

Motori A Combustione Interna E Propulsori Ibridi M

**Sviluppo di algoritmi per l'analisi semi-automatica della
combustione in motori turbo ad accensione comandata**

CANDIDATO:

Giorgio Angelini

RELATORE:

Prof. Nicolò Cavina

CORRELATORI:

Prof. Davide Moro

Prof. Enrico Corti

Ing. Nahuel Rojo

Anno Accademico 2015/2016

Sessione III

Sommario

Introduzione.....	3
1 La combustione e la detonazione.....	5
1.1 La combustione nei motori ad accensione comandata	5
1.2 La detonazione	6
1.2.1 Effetti della detonazione.....	8
1.2.2 Analisi della detonazione	10
2 Il progetto di ricerca	13
2.1 Indice di danno.....	13
2.2 Dati da trattare.....	16
2.3 Il motore	18
2.4 La sala prove	19
3 Metodologia proposta	22
3.1 Operazioni preliminari	25
3.1.1 Ritaglio file indicating e overlap	25
3.1.2 Conversione file bin	25
3.1.3 Calcolo indici per la sincronizzazione.....	25
3.1.4 Calcolo CycleData con HeatIT-off.....	25
3.2 Sincronizzazione	26
3.2.1 Sincronizzazione CycleData e CustomData.....	26
3.2.2 Sincronizzazione CustomData e dati INCA	28
3.2.3 Sincronizzazione dati INCA e banco	30
3.2.4 Output della sincronizzazione	31
3.3 Interfaccia grafica per calcolo indici	34
4 Il segnale di pressione.....	45
4.1 Sensori e catena di misura.....	45
4.2 Acquisizione in sala prove	49
4.3 Trattamento del segnale di pressione	49

4.3.1 Recupero della componente media.....	49
4.3.2 Conversione volt bar	51
4.3.3 Generazione del riferimento angolare	51
4.4 Utilizzo del segnale di pressione per il calcolo di indici.....	52
4.4.1 PMI.....	52
4.4.2 Pmax.....	54
4.4.3 Temperatura	54
4.4.4 Indici di detonazione	54
4.4.5 Legge di rilascio calore	56
5 Il segnale ION.....	64
5.1 Trattamento del segnale ION	66
5.2 Utilizzo del segnale ION.....	67
Conclusioni e sviluppi futuri	69
Bibliografia.....	70

Introduzione

Negli ultimi anni le case automobilistiche hanno effettuato ingenti investimenti nella ricerca in campo motoristico, perseguendo l'obiettivo fondamentale di aumentare l'efficienza dei motori. La spinta all'innovazione è arrivata sia dalle richieste dei clienti, sia dall'adozione di normative sempre più stringenti sulle emissioni e sui consumi.

In questo senso l'analisi della combustione ricopre un ruolo centrale. La combustione è infatti il processo chimico-fisico più importante che avviene all'interno di un motore a combustione interna. Comprendere come essa si sviluppi, riuscire ad analizzarla e controllarla il più possibile significa in ultima analisi poter controllare le prestazioni e l'efficienza del motore.

Ma l'analisi della combustione ha un ruolo importante anche nell'ottica di proteggere il motore da possibili danneggiamenti. Bisogna considerare infatti che i motori di nuova concezione sono generalmente "spinti", cioè hanno una elevata potenza specifica. La maggior parte delle innovazioni tecniche degli ultimi anni va infatti nella direzione dell'aumento della potenza specifica, che permette una riduzione della cilindrata (*downsizing*), e quindi in linea di massima del consumo. Ma motori ad accensione comandata di questo tipo avranno in generale livelli di pressione e temperatura particolarmente elevati, e quindi sono più facilmente soggetti all'insorgenza della detonazione, fenomeno in linea di massima da evitare, perché può danneggiare il motore. Lo studio e l'analisi della combustione risultano quindi ancora più importanti nei motori ad alta potenza specifica.

Al giorno d'oggi, dunque, la progettazione, la realizzazione e la calibrazione di un motore non possono prescindere dall'analisi della combustione.

Il mio lavoro di tesi ha riguardato lo sviluppo di algoritmi in ambiente Matlab per l'analisi della combustione di motori ad accensione comandata.

La prima parte di questa tesi è dedicata ad una introduzione teorica sulla combustione e sulla detonazione nei motori ad accensione comandata.

Successivamente presenterò il progetto di ricerca a cui la mia tesi si è affiancata, descrivendo i dati a disposizione e come questi siano stati ottenuti.

Passerò poi a presentare i risultati del mio lavoro di tesi. Parlerò della metodologia proposta per l'analisi dei dati e spiegherò il funzionamento dei codici che ho realizzato, e sulla base di quali idee li abbia sviluppati.

Nell'ultimo capitolo parlerò invece di come vengano calcolati nei codici gli indici che descrivono la combustione, includendo anche le nozioni teoriche su cui si basano tutte le elaborazioni.

1 La combustione e la detonazione

1.1 La combustione nei motori ad accensione comandata

La combustione è una reazione chimica che comporta l'ossidazione di un combustibile da parte di un comburente (che in genere è rappresentato dall'ossigeno presente nell'aria).

Nei motori a combustione interna la combustione della miscela causa un aumento di pressione che, agendo sulla superficie del pistone, genera la spinta sullo stesso.

I motori ad accensione comandata devono il loro nome al fatto che la combustione deve partire nel punto voluto (ossia dalla candela) e nell'istante desiderato (ossia quando si decide di far arrivare il picco di tensione alla candela).

La combustione viene innescata mediante un piccolo arco elettrico, fatto scoccare fra gli elettrodi della candela.

La fiamma (che in generale nei motori ad accensione comandata con iniezione indiretta è di tipo premiscelato) si propaga quindi nella camera di combustione attraverso un sottile fronte di fiamma, che separa la zona dove è già avvenuta l'ossidazione da quella dove ancora non è avvenuta.

La combustione avanza in virtù del fatto che la zona con i gas bruciati, ad alta temperatura (ordine di grandezza 2000-2500 °K), cede calore attraverso il fronte di fiamma alla zona non ancora ossidata a bassa temperatura (1000 °K), creando così le condizioni per l'autosostentamento della combustione.

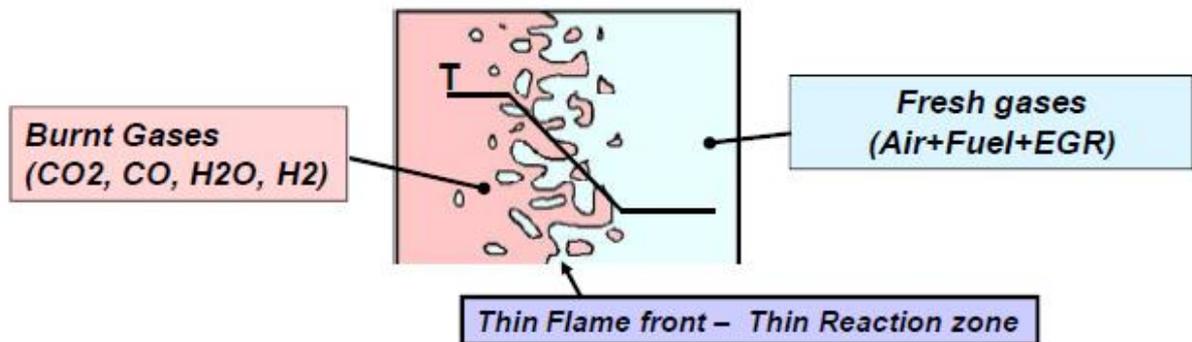


Figura 1.1

1.2 La detonazione

Con l'avanzare della combustione aumentano la pressione e la temperatura in camera. Ebbene, può capitare che alcune porzioni di miscela si trovino in condizioni di pressione e temperatura tali per cui possa avvenire l'autoaccensione.

Questo fenomeno va sotto il nome di detonazione.

La detonazione è dunque un fenomeno di combustione anomala, per cui si ha l'autoaccensione di una porzione di miscela, prima dell'arrivo del fronte di fiamma. È bene sottolineare che col termine *detonazione* si indica l'autoaccensione che avviene eventualmente *dopo* lo scoccare della scintilla: esiste infatti anche un altro fenomeno (molto più raro) di combustione anomala, chiamato pre-accensione, in cui la miscela autoaccende prima dello scoccare della scintilla.

Dal momento in cui si realizzano le condizioni per l'autoaccensione trascorre poi un certo tempo, detto "tempo di ritardo", prima che la combustione detonante abbia effettivamente inizio.

Il corretto funzionamento della combustione viene in definitiva delegato ad una "gara di velocità" fra la combustione normale, caratterizzata dall'avanzamento graduale del fronte di fiamma, e quella detonante.

Di conseguenza, è evidente che per evitare l'insorgenza della detonazione esistono due strade fondamentali: aumentare il tempo di ritardo, sia agendo sulla chimica del combustibile (numero di ottani) che soprattutto controllando le condizioni di pressione e temperatura in camera (carico, anticipo di accensione), e/o aumentare la velocità di combustione. Di questa seconda possibilità parlerò più diffusamente.

La velocità di combustione può essere governata agendo su vari parametri.

In primis, il titolo della miscela: la velocità di combustione è massima per miscele leggermente grasse, e decade velocemente con miscele magre o molto grasse. Da qui deriva anche il fatto che i motori ad accensione comandata non possano essere regolati agendo sulla qualità della miscela, pena l'insorgenza della detonazione.

Enorme importanza ha anche la fluidodinamica interna del motore. Per moti di tipo turbolento la velocità di combustione è sensibilmente più alta rispetto al moto laminare. Da questo discende l'importanza di una corretta progettazione fluidodinamica dei condotti di aspirazione e della camera di combustione stessa.

La figura 1.2 riporta l'andamento della velocità di combustione al variare del titolo. La linea continua indica l'andamento della velocità di combustione in un generico motore,

in cui il moto è fondamentalmente turbolento, mentre la linea tratteggiata rappresenta una condizione di moto laminare, realizzata in laboratorio.

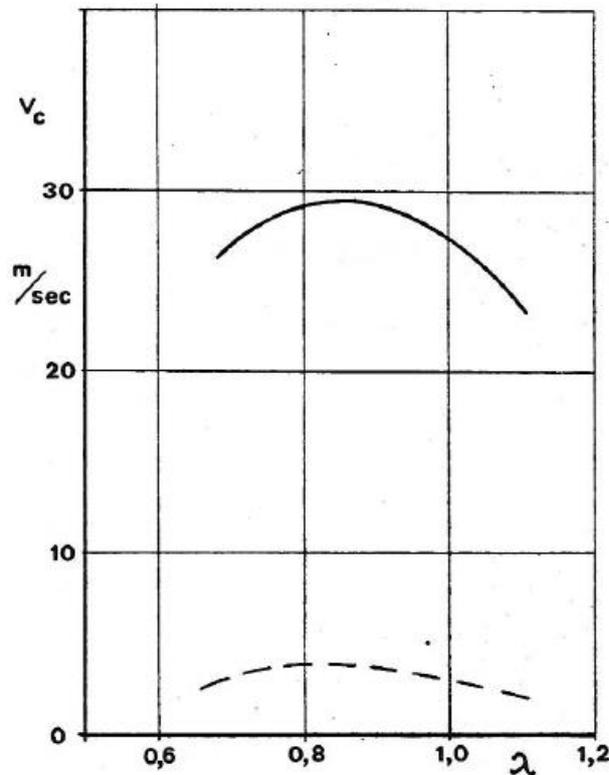


Figura 1.2

In definitiva, saranno critiche dunque tutte le condizioni che comportano elevate pressione e temperatura in camera: alto carico, elevato anticipo, elevata pressione di sovralimentazione, alte temperatura e pressione ambientale...

Altrettanto favorevoli all'insorgenza della detonazione sono tutte le condizioni che comportano bassa velocità del fronte di fiamma: basso numero di giri (poca turbolenza in camera), titolo troppo magro o troppo grasso...

La detonazione limita superiormente le prestazioni ottenibili dai motori ad accensione comandata, in termini di potenza specifica. Aumentare la potenza specifica significa infatti in generale creare più elevati livelli di pressione e temperatura in camera. L'insorgenza della detonazione limita anche la cilindrata unitaria nei motori ad accensione comandata, e in particolare l'alesaggio, perché il fronte di fiamma si troverebbe a dover percorrere distanze troppo "lunghe". È pratica comune ormai limitare ogni cilindro a circa 600 cm^3 .

La detonazione limita anche il rendimento termodinamico ottenibile dal motore. Esiste infatti una relazione che mostra che il rendimento termodinamico del ciclo Otto dipende dal rapporto di compressione. Ma aumentando il rapporto di compressione aumentano in generale i livelli di pressione e temperatura in camera, e quindi è più facile incorrere nella detonazione.

1.2.1 Effetti della detonazione

La porzione di miscela sottoposta a detonazione si accende “tutta insieme” (da questo fatto deriva anche il nome *detonazione*, che rimanda ad una “esplosione”), e non gradualmente (come avviene con il normale avanzamento del fronte di fiamma). A questo tipo di combustione sono associati enormi picchi di pressione e temperatura, che possono addirittura danneggiare il motore, anche perché localizzati sempre nelle stesse parti della camera di combustione, ossia generalmente nelle zone più lontane dalla candela.

Nei motori con candela in posizione centrale (scelta ormai obbligata e dettata dal bisogno di ridurre il percorso massimo del fronte di fiamma) le zone interessate sono quindi quelle vicine alla superficie del cilindro e tendenzialmente verso le valvole di scarico, in cui la presenza di punti caldi crea le condizioni maggiormente favorevoli all’autoaccensione.

La detonazione erode parti del pistone, con modalità simile alla cavitazione, e contribuisce ad un generale rammollimento del materiale, per via delle elevate temperature.

Sulle modalità di danneggiamento causate dalla detonazione tornerò in modo più approfondito nel capitolo dedicato alla presentazione del progetto di ricerca.

La figura 1.3 mostra alcuni danni indotti da combustioni detonanti.

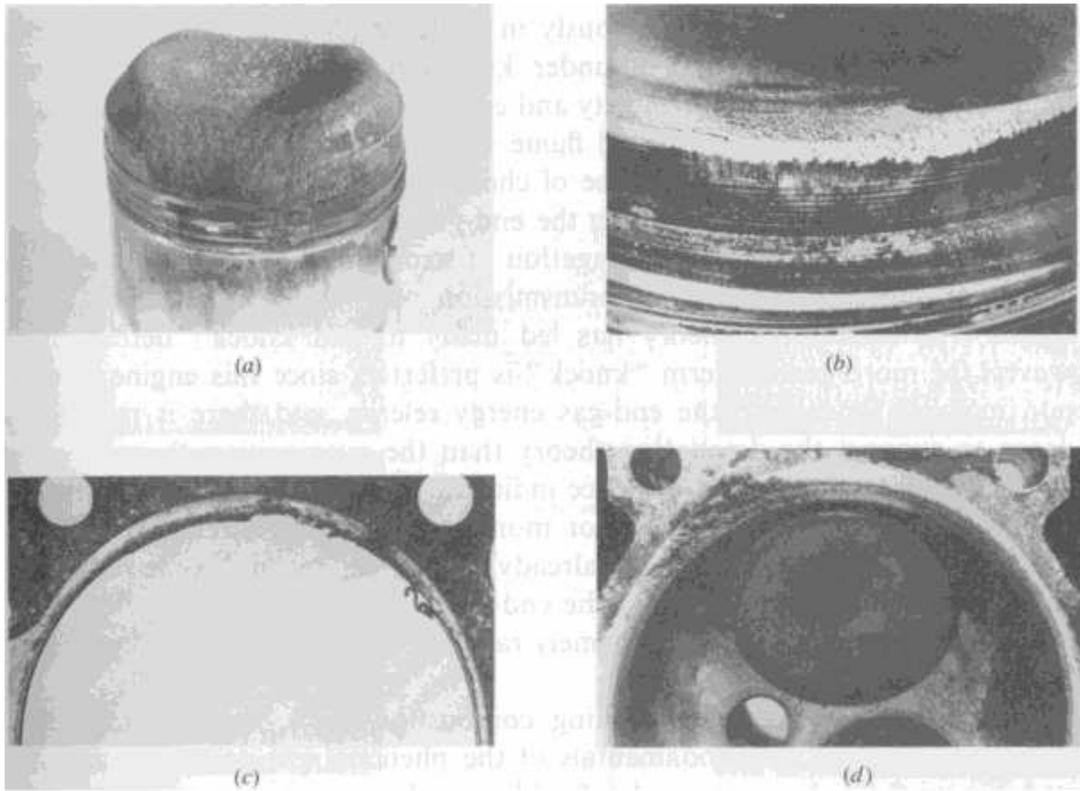


FIGURE 9-61

Examples of component damage from abnormal engine combustion. (a) Piston holing by preignition;⁸³ (b) piston crown erosion after 10 hours of high-speed knocking;⁸² (c) cylinder head gasket splitting failure due to heavy knock;⁸³ (d) erosion of aluminum cylinder head along the top of the cylinder liner due to heavy knock.⁸³

Figura 1.3

Finora abbiamo parlato della detonazione come una condizione da evitare, perché causa di possibile danneggiamento del motore.

In realtà, le moderne centraline di controllo motore operano consentendo un certo livello di detonazione (comunque estremamente ridotto), nell'ottica di aumentare la potenza specifica e ridurre i consumi.

In particolare, la ECU ha una logica di funzionamento che prende in considerazione il valore statistico (percentile in genere) di un indice di detonazione e corregge l'anticipo di accensione (lo riduce) qualora tale valore superi un determinato livello.

La detonazione, come spiegherò in modo approfondito più avanti, può essere "osservata" e "misurata" sul segnale di pressione in camera di combustione. Ma in vettura non si installano normalmente i sensori di pressione perché essi sono molto costosi ed hanno anche una durata limitata. L'indicazione sulla detonazione in vettura

può provenire allora da altri segnali: accelerometrico, acustico o di corrente di ionizzazione.

In fase di calibrazione si definiscono degli indici basati sui segnali disponibili in vettura in modo che ci sia alta correlazione con gli indici basati sul segnale di pressione.

La soglia di detonazione “accettata” è in genere stabilita o concordata col produttore del motore (o dei pistoni) e viene applicata in fase di calibrazione; in generale dipende anche dall’applicazione specifica del motore.

L’ideale sarebbe trovare il livello di detonazione tale per cui il motore ha la durata stabilita a progetto, con coefficiente di sicurezza unitario, e che allo stesso tempo consenta di ottenere le massime prestazioni.

1.2.2 Analisi della detonazione

Nell’ambito dell’analisi della combustione, la detonazione è uno dei fenomeni più studiati.

Essa è “visibile” sul segnale di pressione, su cui produce delle oscillazioni ad alta frequenza.

La detonazione genera infatti delle onde di pressione (picchi e depressioni) che si riflettono all’interno della camera di combustione alla velocità del suono. Le suddette onde hanno frequenze tipicamente comprese tra 5 e 20 kHz (calcolabili noti l’alesaggio e stimando la temperatura, da cui dipende la velocità di propagazione delle onde stesse). Ebbene, in questo *range* si trovano anche alcune delle frequenze proprie della camera di combustione, che quindi vengono eccitate: da qui la comparsa delle oscillazioni ad alta frequenza sul segnale di pressione.

L’indice più utilizzato per caratterizzare la detonazione, il MAPO, è proprio definito come il massimo valore delle oscillazioni ad alta frequenza della pressione in camera di combustione di un singolo ciclo.

Da notare che non esiste un valore di MAPO (o di qualunque altro indice) che distingua nettamente tra combustione normale o detonante. Questo anche perché, per esempio, con l’aumentare dei giri la combustione diventa più ruvida, più veloce. In altre parole, ad alto regime di rotazione le alte frequenze del segnale di pressione hanno un contenuto energetico più elevato (cioè la pressione varia più velocemente).

Per esempio, a 6000 giri al minuto corrisponde una frequenza di ciclo di 50 Hz: se di questa consideriamo la 50esima armonica, essa avrà una frequenza di 2,5 kHz (cioè le alte frequenze hanno contenuto energetico anche in presenza di combustione normale). Esiste una semplice regola empirica che definisce un MAPO di soglia oltre al quale la combustione può essere considerata pericolosamente detonante:

$$\text{MAPO}_{\text{soglia}} [\text{bar}] = \text{rpm} / 1000;$$

In associazione ad eventi detonanti si nota anche un aumento del calore disperso alle pareti, che si può stimare attraverso l'analisi delle grandezze termodinamiche (in realtà anche questo tipo di analisi, come vedremo più avanti, si basa sul segnale di pressione). Associata all'aumento delle perdite a parete può esserci la diminuzione dell'efficienza complessiva.

Sempre in ottica analisi, è interessante notare che se un motore viene portato in condizioni detonanti, aumenta non solo l'indice tipico (per esempio il MAPO), ma anche la deviazione standard dello stesso. Questo avviene perché la detonazione non è un fenomeno deterministico. L'insorgenza o meno di una combustione detonante dipende infatti pesantemente da come avviene la prima fase della combustione, che è molto poco ripetibile. Se la combustione parte troppo lentamente, è difficile aumentarne la velocità successivamente: da qui il maggior pericolo di incorrere nella detonazione.

L'analisi della detonazione è quindi di tipo statistico. In linea generale, un motore che non detona avrà una distribuzione del MAPO di tipo gaussiano, con una "campana" ben definita, e molto stretta. In presenza di detonazione aumentano la deviazione standard e la media della distribuzione, che tende ad essere log-normale.

2 Il progetto di ricerca

La mia attività di tesi si è inserita all'interno di un progetto di ricerca dell'Università di Bologna svolto in collaborazione con Ferrari S.p.A.

Il progetto mira ad ottenere una migliore comprensione del fenomeno della detonazione, ed in particolare dei danni che essa induce sul motore.

Si è supposto, e verificato con prove sperimentali, che esistano due tipologie fondamentali di danno indotto dalla detonazione: il danno di origine meccanica e quello di origine termica.

Il danno di tipo meccanico è quello a cui si associa “classicamente” la detonazione. La propagazione delle onde di pressione all'interno della camera di combustione può causare danni gravi, visibili anche ad occhio nudo, principalmente sui pistoni e sulle fasce elastiche, insieme a variazioni di rugosità.

Meno rappresentato in letteratura, ma non per questo meno importante, è il danneggiamento per via termica. Si è visto che la detonazione provoca un aumento del calore scambiato a parete, e dunque anche del calore che investe il pistone. Ma ad una maggiore temperatura del pistone corrisponde un più rapido invecchiamento del materiale dello stesso, secondo le curve di invecchiamento caratteristiche del materiale. In particolare si nota (e si misura sperimentalmente) il calo della durezza residua del materiale dei pistoni.

Le due modalità di danneggiamento, inoltre, interagiscono tra loro, nel senso che un invecchiamento precoce del materiale (danno di tipo termico) rende lo stesso più soggetto allo stress meccanico dovuto alle onde di pressione.

2.1 Indice di danno

L'idea è allora quella di costruire un indice che rappresenti il livello di danneggiamento del materiale indotto dalla detonazione, che ovviamente tenga conto di entrambe le modalità sopra esposte.

Il suddetto indice potrà poi essere inserito, una volta che esso sia stato ben tarato, nella centralina di controllo motore. La ECU potrebbe così usare l'informazione sullo stato dei pistoni per chiudere il loop di controllo dell'anticipo.

I sistemi di controllo dell'anticipo, ad oggi, si basano fondamentalmente su *mappe* che vengono inserite in centralina in fase di calibrazione del motore. È evidente che un controllo di questo tipo richiede tanto tempo (e quindi tante risorse economiche) per la fase di calibrazione, e non permette di ottenere le massime prestazioni. Il motore infatti è soggetto ad invecchiamento, e inoltre ogni motore che esce dalla catena di montaggio non è perfettamente uguale a quello usato per la calibrazione...

Più recentemente sono stati introdotti sistemi di controllo in *closed-loop*. Nel controllo in catena chiusa vengono usati sensori ION, accelerometri o sensori acustici per chiudere la catena: tuttavia la retroazione funziona solo in ottica di *protezione* dalla detonazione, ha cioè solo il "permesso" di togliere gradi di anticipo.

L'idea è allora quella di chiudere il loop di controllo dell'anticipo con target "velocità di danneggiamento" del motore, misurata basandosi sull'indice di cui sto parlando, realizzando quindi un controllo aggressivo in *closed-loop*.

Se la centralina rileva che la velocità di danneggiamento è troppo elevata comanderà in linea generale un anticipo meno aggressivo, abbassando di fatto il livello di detonazione consentita, diminuendo le prestazioni. Viceversa avverrà nel caso che la centralina rilevi dall'indice che il motore è meno danneggiato del previsto. Un controllo di questo tipo rappresenta di fatto un possibile margine di guadagno sulle prestazioni del motore, da cui il forte interesse industriale.

Ma vediamo nel dettaglio come andrebbe implementato l'utilizzo dell'indice in vettura.

- Si stabilisce in fase di progetto la durata prevista del motore (in termini di ore di funzionamento o di chilometri percorsi), in base all'applicazione, e il livello massimo di danno tollerato (cioè dell'indice).
- Si determina poi quale sia il valor medio dell'indice di danno che permette il raggiungimento del target (pendenza dell'indice cumulato), che rappresenta il target del sistema di controllo.
- Il carico termico, informazione che in generale sarà necessaria, può essere modellato per esempio a partire dalla massa d'aria aspirata e dalla posizione del picco del segnale ION.

Allo stato attuale di avanzamento del progetto, ci si sta concentrando su due problematiche fondamentali per irrobustire i risultati ottenuti finora:

- Ottenere un modello attendibile della temperatura del cielo del pistone
- Cercare di definire quantitativamente la relazione esistente tra livello di detonazione e scambio di calore a parete.

2.2 Dati da trattare

La natura del progetto di ricerca, che prevede di analizzare i danni indotti sui pistoni dalla detonazione, richiede tante ore di funzionamento del motore, e quindi di prove sperimentali al banco, e quindi tante ore di acquisizione dati.

I dati che si acquisiscono in sala prove sono fondamentalmente di tre tipi: il segnale indicating, i dati INCA e i dati banco.

I dati indicating contengono l'andamento della pressione in camera di combustione all'interno del ciclo, per ogni ciclo e per ogni cilindro. In generale sono le informazioni più preziose e dense di contenuto. Come spiegato più approfonditamente nel capitolo relativo al trattamento del segnale di pressione, sulla base di esso si possono calcolare varie grandezze termodinamiche, si può descrivere l'andamento della combustione e si può "osservare" ed analizzare la detonazione. Il knock, infatti, si manifesta sul segnale di pressione come un'oscillazione ad alta frequenza.

I dati banco e INCA descrivono le condizioni al contorno e di controllo del motore, come per esempio la velocità di rotazione, la percentuale di apertura farfalla, la pressione e la temperatura del collettore, il titolo allo scarico, l'anticipo attuato, la coppia e la potenza erogate. Sull'acquisizione INCA, per la specifica configurazione della cella motore, sono presenti anche alcune grandezze che derivano dalla traccia di pressione, in virtù della comunicazione (basata su protocollo XCP) tra il sistema indicating e INCA.

Alcune informazioni provengono dalla centralina e quindi sono normalmente acquisite dal software INCA, altre sono disponibili solamente nel contesto di laboratorio, attraverso la strumentazione di cella, e sono quindi acquisite dal sistema "banco".

Se si considera ora che il campionamento del segnale di pressione avviene ad alta frequenza (200 kHz), e che una prova genera in uscita un unico file *streaming* (con tutti i cicli uno di fila all'altro), è facile immaginare che il file indicating possa avere dimensioni notevoli.

I dati INCA e banco vengono invece acquisiti con frequenze molto più basse, e quindi hanno dimensioni più contenute.

Proprio dalla dimensione dei file indicating nasce l'esigenza di effettuare un pre-processing del segnale e di avere uno strumento comodo, affidabile e soprattutto efficiente per il calcolo degli indici basati sul segnale di pressione.

2.3 Il motore

Il motore oggetto dello studio è il propulsore Ferrari F154BB, con 8 cilindri disposti a V. Si tratta di un motore ad elevate prestazioni, destinato a vetture sportive. La potenza di picco è pari a 560 CV (412 KW), per una cilindrata di 3.8 litri. Per una casa come la Ferrari, i cui motori hanno cilindrata che normalmente superano i 5 litri, si può parlare propriamente di strategia di *downsizing*. Il propulsore ha infatti una elevatissima potenza specifica, che supera i 145 CV/litro. Un valore tanto elevato è ottenuto anche grazie all'utilizzo della sovralimentazione, con un gruppo turbocompressore per ogni bancata di cilindri.

È evidente che per un motore del genere lo studio della detonazione è ancora più importante, in virtù della elevata potenza specifica.

Il propulsore presenta molte soluzioni tecniche all'avanguardia, come l'iniezione diretta di benzina e la presenza del sistema di analisi della corrente di ionizzazione.



Figura 2.1. Il motore F154BB

2.4 La sala prove

La sala prove dell'Università di Bologna si trova nei laboratori DIN di via Terracini.

Le prove sul motore vengono svolte dalla sala di controllo, dove sono installati i pc che permettono il controllo della cella motore e di tutti i sistemi ausiliari, nonché l'acquisizione di tutti i dati disponibili e ritenuti utili.

In particolare un pc presenta il software di controllo banco, che ha la funzione fondamentale di regolare il freno dinamometrico ed il carico del motore, e permette anche la visualizzazione di alcune grandezze.

Un altro pc mostra invece il sistema di analisi della combustione real-time basato sui dati indicanti provenienti da OBI, visualizzando in tempo reale le curve di pressione ed altri parametri.

Un terzo computer consente invece di interfacciarsi con la ECU, tramite il programma INCA: quest'ultimo comunica con la centralina attraverso un modulo aggiuntivo ETK (specifico per l'attività in sala prove) e permette all'operatore di visualizzare e modificare le variabili della ECU.

In sala sono presenti tantissimi sensori, per esempio sensori di temperatura, pressione, umidità... Elemento fondamentale della cella è il freno dinamometrico a correnti parassite, che consente di misurare la coppia fornita dal motore. Sia il freno che il motore sono raffreddati ad acqua, e l'acqua viene a sua volta raffreddata per mezzo di una apposita torre di raffreddamento esterna.

Numerose sono le differenze tra l'installazione di un motore a banco e in vettura.

Innanzitutto, sul banco prova possono non essere presenti (come nel caso specifico) il cambio e tutta la trasmissione, che vengono "sostituiti" dal freno dinamometrico.

Inoltre in cella non è presente il vento di corsa, e quindi il sistema di raffreddamento del motore è completamente diverso da quello usato in vettura. Gli intercooler a valle dei compressori sono degli scambiatori aria/acqua (e non aria/aria, come in vettura), e manca il radiatore.

Inoltre sulla linea di scarico non è presente la sonda λ a valle catalizzatore, che serve in vettura per la diagnosi del funzionamento del catalizzatore.

Durante le prove il motore in esame viene generalmente utilizzato con una sola bancata attiva, con 200 KW di potenza (a parte nella fase di riscaldamento), perché la sala non è dimensionata per sopportarne la piena potenza. Infatti, una regola empirica indica

che per ogni KW di potenza meccanica prodotto dal motore debbano essere smaltiti 3 KW totali di potenza termica (1 KW prodotto dal freno a correnti parassite, 1 KW asportato da acqua di raffreddamento e olio, 1 KW nell'entalpia dei gas di scarico). Allora, se si usasse il motore alla piena potenza, di 400 KW, bisognerebbe smaltire 1200 KW, mentre la torre di raffreddamento è dimensionata per soli 700 KW.



Figura 2.2. Installazione del motore a banco.



Figura 2.3. Collegamento del motore al freno dinamometrico.

3 Metodologia proposta

Facendo riferimento allo schema della pagina successiva, presento la metodologia proposta per l'analisi dei dati.

L'attività sperimentale al banco prova fornisce in uscita i file con i segnali indicating, banco e INCA.

Il file indicating, in formato bin, come abbiamo detto contiene il segnale di pressione acquisito ad alta frequenza (200 kHz), e può essere quindi di grande dimensione.

I programmi a disposizione per il post-processing non possono trattare file troppo grandi, quindi occorre procedere alla divisione del file streaming indicating in porzioni più piccole.

La dimensione di ogni spezzone di file deve essere tale da garantire la "trattabilità" con i programmi di post-processing. Per quanto riguarda HeatIT-off, si avranno file relativamente "grandi" (circa 1 GB) da utilizzare per il calcolo degli indici (CycleData). I ritagli destinati ad essere convertiti in formato mat, per l'elaborazione con Matlab, sono invece più piccoli, circa 200 MB: di essi useremo il segnale in base tempo.

I file mat vengono quindi elaborati da una apposita interfaccia grafica, sviluppata in ambiente Matlab, che effettua il calcolo di alcuni indici (i CustomData).

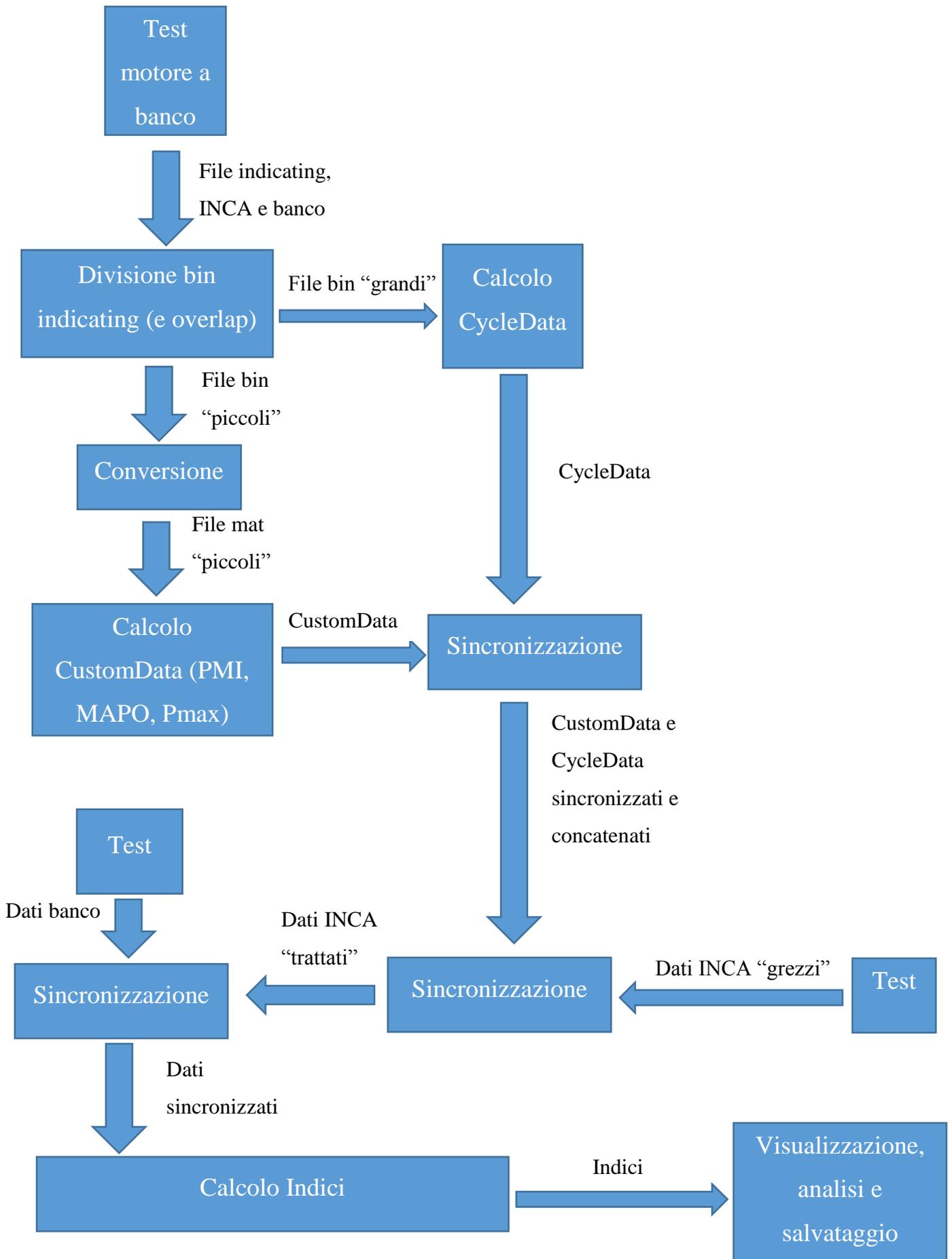
In questa prima fase vengono usate solo le function che calcolano il MAPO, la pressione massima e la PMI: nel file indicating mancano infatti alcune informazioni (anticipo, massa d'aria, lambda, consumo...) che sono necessarie per il calcolo di altri indici.

Le informazioni mancanti sui file mat sono presenti invece nei file di output dell'acquisizione banco e INCA.

Prima di poter accedere a questi dati occorre tuttavia sincronizzare tra di loro i segnali, le cui acquisizioni, in generale, possono partire ad istanti diversi ed avere anche durate diverse.

Inizialmente vengono sincronizzati i CycleData e i CustomData, utilizzando la PMI e il MAPO.

Successivamente vengono sincronizzati i CustomData con i dati INCA, utilizzando la PMI, il MAPO e la pressione massima.



Infine vengono sincronizzati i dati banco e i dati INCA, facendo riferimento alla temperatura a monte farfalla.

Ora che tutti i dati sono sincronizzati, e abbiamo dunque accesso alle informazioni contenute nei dati INCA e banco, possiamo lanciare nuovamente l'interfaccia grafica per calcolare tutti gli indici desiderati.

Gli indici possono poi essere visualizzati ed analizzati mediante un'altra interfaccia grafica, che permette tra le altre cose di impostare interattivamente delle condizioni logiche (per esempio, visualizzare tutti i cicli con un MAPO maggiore di una certa soglia).

La mia attività di tesi ha riguardato principalmente lo sviluppo del codice che effettua la sincronizzazione e dell'interfaccia grafica per il calcolo degli indici.

3.1 Operazioni preliminari

3.1.1 Ritaglio file indicating e overlap

Abbiamo detto che il file indicating streaming deve essere diviso in porzioni più piccole. Questa operazione viene effettuata da un apposito eseguibile, predisposto per la suddivisione dei file bin.

L'eseguibile che effettua il ritaglio dei file genera anche la sovrapposizione (“overlap”) di alcuni cicli motore. L'overlap si rende necessario perché il ritaglio viene fatto su base tempo, e non su base ciclo, per cui può accadere che alcuni cicli rimangano incompleti (e quindi non avrebbe senso analizzarli). Con l'overlap si “allunga” di fatto ogni file ritagliato, aggiungendo una porzione del file successivo: in questo modo si è sicuri di non dover scartare nessun ciclo (perché incompleto) nella fase di post-processing. L'overlap si effettua sia sui file destinati ad essere trattati con Matlab, sia su quelli destinati al calcolo dei CycleData con HeatIT-off.

3.1.2 Conversione file bin

Usando l'apposita funzione di HeatIT-off si convertono i ritagli di file bin “piccoli” in formato mat, adatto ad essere trattato con Matlab.

3.1.3 Calcolo indici per la sincronizzazione

In questa fase vengono calcolati tramite l'interfaccia (che presenterò in modo dettagliato più avanti) solo gli indici che vengono usati dal codice di sincronizzazione, e dunque PMI, MAPO e pressione massima. Ovviamente deve essere selezionata sull'interfaccia la modalità che provvede al salvataggio degli indici.

3.1.4 Calcolo CycleData con HeatIT-off

Usando HeatIT-off si calcolano i CycleData, cioè gli indici ciclo per ciclo che verranno utilizzati anche dall'algoritmo di sincronizzazione.

3.2 Sincronizzazione

I segnali provenienti da sorgenti diverse devono essere sincronizzati tra di loro, sia perché possono avere durate diverse (in genere le acquisizioni banco e INCA iniziano prima e finiscono dopo l'acquisizione indicating), sia a causa dell'operazione di ritaglio a cui vanno sottoposti i dati indicating.

Nella presentazione della metodologia ho distinto tre fasi diverse nell'operazione di sincronizzazione dei dati. In realtà, tutte le operazioni vengono eseguite da un unico codice di sincronizzazione.

Abbiamo deciso di individuare una struttura fissa per la cartella che contiene i dati relativi a ciascuna prova. In particolare, nella generica cartella "Test" dovremo inserire:

- Una cartella "banco", contenente l'acquisizione banco.
- Una cartella "inca", contenente l'acquisizione INCA.
- Una cartella "divisi_CycleDomain", che contiene i CycleData calcolati con HeatIT-off.
- Una cartella "divisi_TimeDomain", che contiene i ritagli "piccoli" di file indicating e la cartella "indici", in cui vengono salvati gli indici calcolati con l'interfaccia.
- Eventualmente il file di configurazione.

In questo modo, il codice chiede all'utente solo di selezionare la cartella contenente i dati relativi ad una certa prova, e poi seleziona autonomamente tutti i file necessari per la sincronizzazione.

3.2.1 Sincronizzazione CycleData e CustomData

La prima fase di sincronizzazione dei dati riguarda i CycleData, cioè gli indici calcolati con HeatIT-off (a partire dai ritagli "grandi" dei file bin), e i CustomData, cioè gli indici calcolati con l'interfaccia grafica (a partire dai ritagli "piccoli" dei file bin).

Tra i CycleData troviamo ad esempio la PMI, il MAPO (questi due sono usati proprio per la sincronizzazione), la pressione massima, il calore netto (CHR), gli angoli che descrivono l'andamento della combustione (MFB)...

CycleData e Customdata, pur derivando entrambi dall'acquisizione indicating, devono essere sincronizzati prima di poter procedere alla loro concatenazione, perché i dati utilizzati hanno dimensioni (in termini di numero di cicli considerati) diverse. Inoltre, alcuni cicli vengono ripetuti (per l'overlap), e in numero non esattamente costante tra un file ed il successivo.

Il codice in questa fase utilizza la PMI e il MAPO per la sincronizzazione.

L'output di questa prima parte di codice sono i CustomData e i CycleData (sotto forma di struttura Matlab) sincronizzati e concatenati, e la struttura RIF_CYCLES. Questa struttura contiene i riferimenti ciclo *globali* (cioè relativi a tutta l'acquisizione) e *locali* (cioè per ogni ritaglio di file) sia per i CustomData che per i CycleData.

I riferimenti *locali* sono indicati sia con il numero del ciclo, sia con il nome del file corrispondente: quest'ultima informazione verrà usata nell'interfaccia grafica per assegnare correttamente i valori delle acquisizioni INCA e banco a ciascun file indicating.

Questa prima parte di codice può presentare una criticità di funzionamento. Nella figura sottostante sono rappresentati i riferimenti ciclo (numeri) globali e locali per i CustomData e i CycleData. In questo set di dati ogni file di CustomData contiene circa 1000 cicli, mentre ogni file di CycleData ne contiene circa 5000.

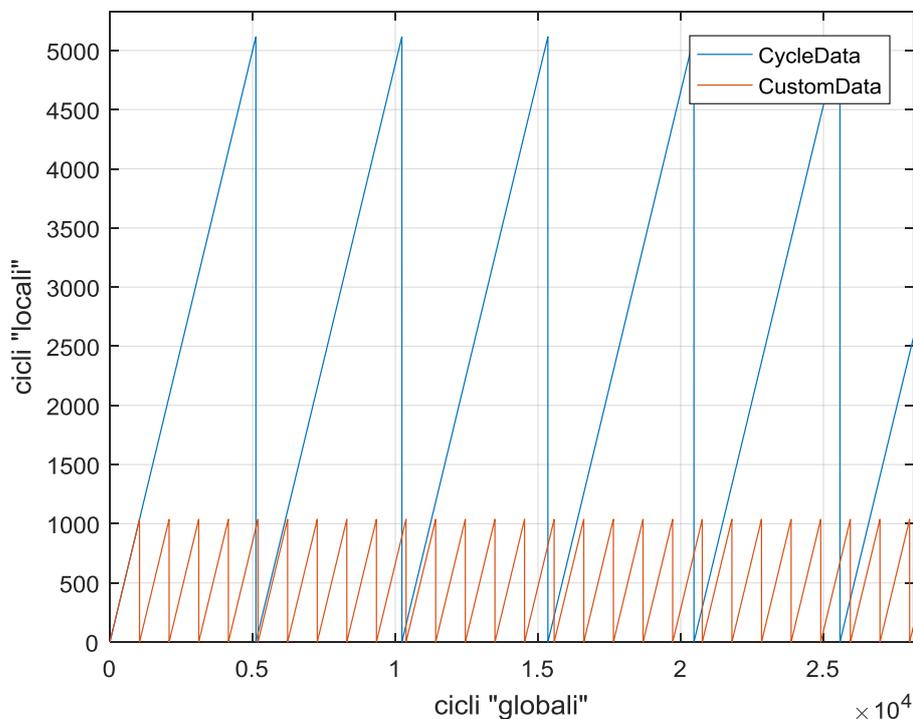


Figura 3.1

Ebbene, quando il codice ha terminato di sincronizzare i cicli su un file di CycleData, carica il file successivo contenuto nella cartella CycleDomain. In questo frangente, però, la sincronizzazione viene fatta considerando solo la “differenza di cicli” tra il corrente file di CustomData e il nuovo file di CycleData: se questa differenza è molto piccola (al limite può essere anche nulla), l’algoritmo di sincronizzazione può sbagliare. Nella figura sottostante riporto due casi in cui la ”differenza di cicli” è sensibilmente diversa.

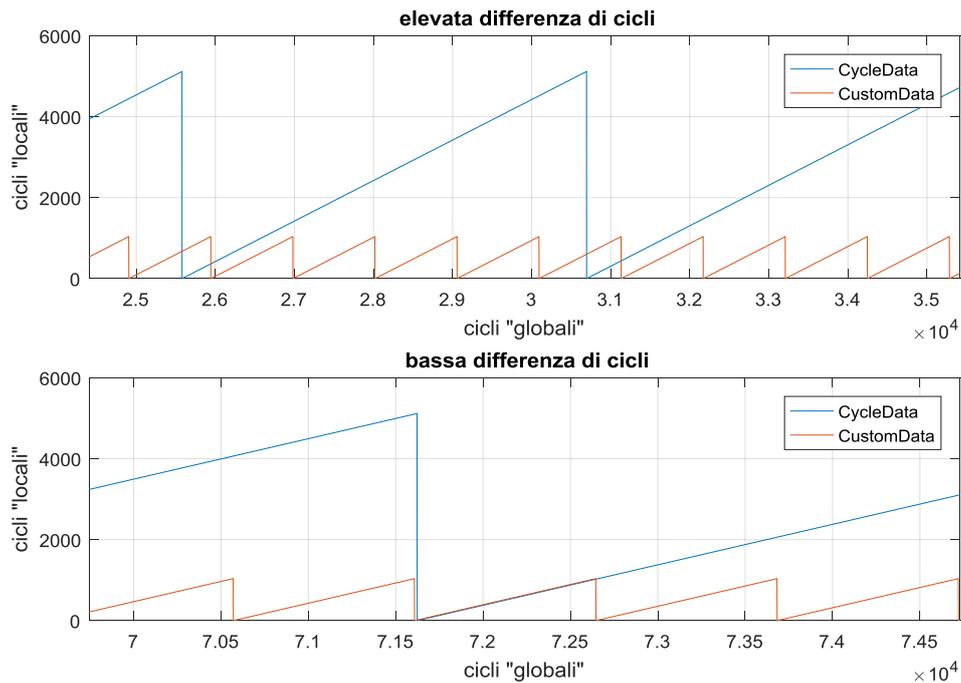


Figura 3.2

Effettuando delle prove abbiamo visto però che in linea generale questo problema si può risolvere a monte, ritagliando i file indicating in modo che non si verifichi la situazione con bassa differenza di cicli.

3.2.2 Sincronizzazione CustomData e dati INCA

La seconda parte dello script di sincronizzazione opera sui CustomData (concatenati nella parte precedente del codice) e sui dati INCA, caricati dalla cartella corrispondente.

Come prima cosa lo script genera un vettore “tempo_ciclo” associato ai dati indicating, calcolato a partire dagli RPM (informazione contenuta nei CycleData).

```
tempo_ciclo = cumsum(120./CYCLEDATA.RPM);
```

L'idea fondamentale è quella di ricampionare il segnale INCA sulla base del vettore "tempo_ciclo", in modo che i due segnali abbiano la stessa frequenza di campionamento. A questo punto si può operare la sincronizzazione, che avviene usando il MAPO, la PMI e la pressione massima. Nell'acquisizione INCA sono infatti presenti, oltre alle grandezze tipiche di centralina, anche i dati sopra menzionati (e altri derivanti dal segnale di pressione), in virtù della comunicazione (basata su protocollo XCP) tra il sistema indicating e INCA.

Il segnale INCA è affetto però da un forte jitter, che rende non accurata la sincronizzazione. Nel codice è implementata allora una strategia che distorce il vettore di interpolazione "tempo_ciclo", con target di massima correlazione.

Ho analizzato un set di dati relativo ad una prova in cui l'anticipo è stato controllato in maniera particolare: ogni 50 cicli si passa a gradino da 6,75° a 12,75° di anticipo. Un tale "salto" facilita l'insorgenza di combustioni detonanti. Ho verificato con questo set di dati che con l'algoritmo che distorce il vettore "tempo_ciclo" si realizza un errore massimo di sei cicli. Questo si può verificare per esempio plottando gli anticipi estratti da INCA e i MAPO contenuti nei CustomData (o nei CycleData).

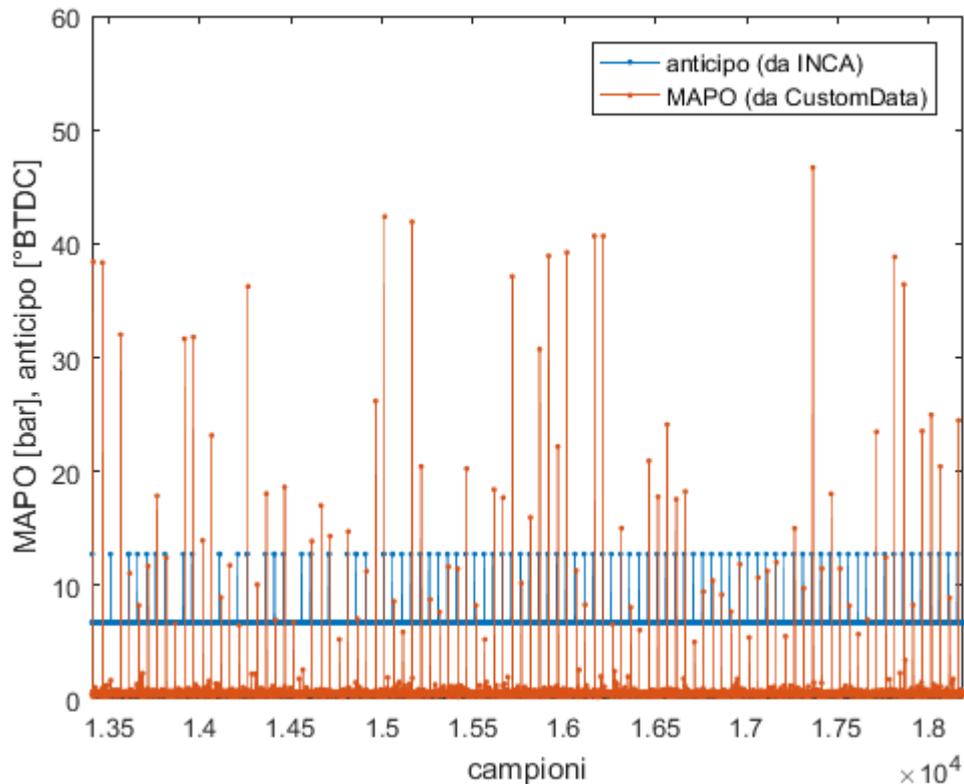


Figura 3.3

Ho provato anche altre strade per compensare gli errori sul segnale INCA: modifica del vettore `time_14` (che è il *raster* temporale con cui vengono calcolate le grandezze indicating), filtraggio sempre di `time_14`, ricostruzione di `time_14` usando uno spazio lineare (comando `linspace`). Tuttavia, di queste soluzioni quella che ha portato al risultato migliore è stata proprio la modifica del vettore “tempo_ciclo”.

3.2.3 Sincronizzazione dati INCA e banco

L’ultima parte dello script di sincronizzazione opera sui dati banco e INCA.

Il dato banco è in formato txt, e quindi una parte di codice è delegata a leggere questo file e renderne disponibile il contenuto nel *workspace* di Matlab.

Il codice riconosce automaticamente quale sia la bancata attiva del motore, e “capisce” quindi quali siano le variabili giuste da usare per la sincronizzazione e da estrarre. Il riconoscimento avviene sulla base delle temperature medie dei collettori di scarico, che come si vede dalla figura 3.4 sono molto diverse tra bancata accesa e bancata spenta (l’algoritmo è quindi “robusto”).

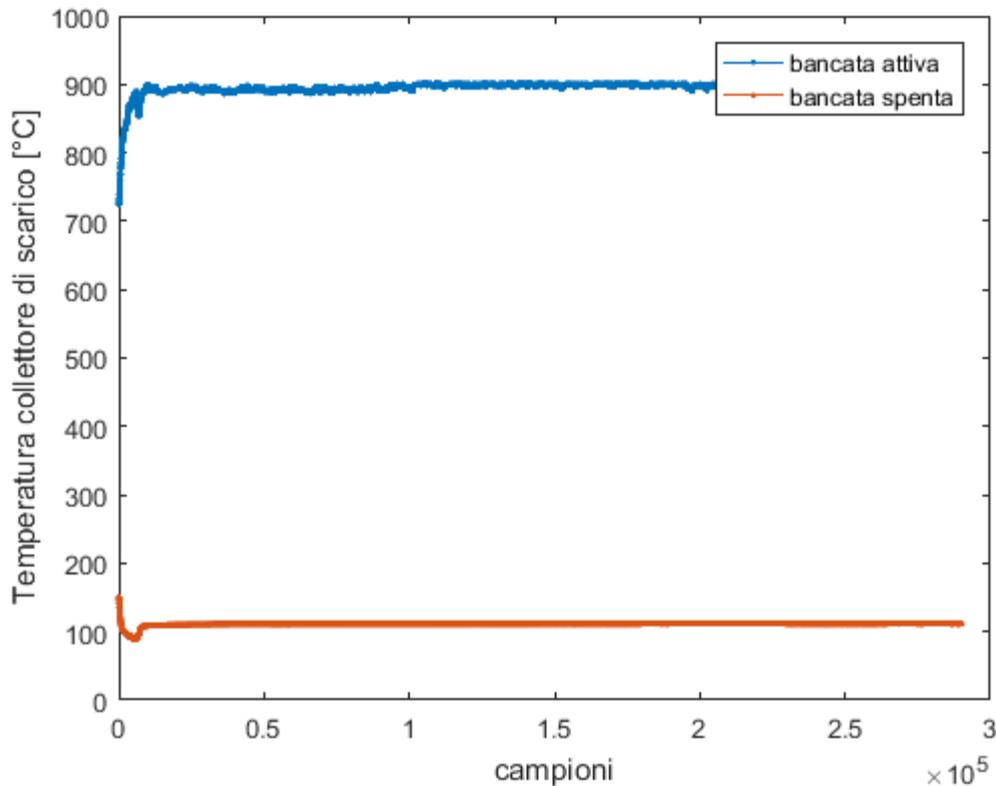


Figura 3.4

Questa fase della sincronizzazione viene realizzata usando il segnale di temperatura a monte farfalla.

Le misure contenute nei dati INCA e banco sono diverse perché fatte in punti vicini e con sensori diversi: il segnale INCA deriva da un sensore “di serie”, mentre il segnale banco deriva da un sensore che fa parte della strumentazione di cella.

Il risultato è in questo caso ottimo, i due segnali sono praticamente in fase.

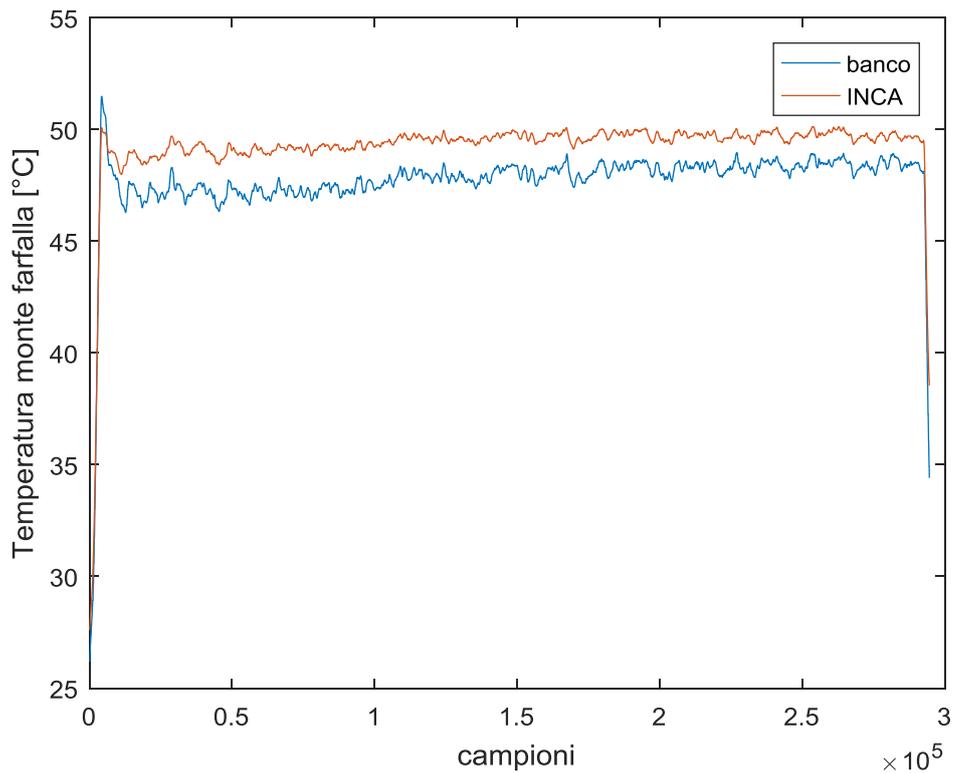


Figura 3.5

3.2.4 Output della sincronizzazione

Il codice di sincronizzazione genera come output un unico file, contenente i Customdata, i CycleData, i riferimenti ciclo, i dati banco e INCA ricampionati. Come ho detto, il codice riconosce autonomamente quale sia la bancata attiva, ed estrae quindi solo i dati relativi ai cilindri funzionanti.

Dai dati INCA preleviamo in particolare gli anticipi, la massa d'aria e il titolo.

Dai dati banco preleviamo invece il consumo e la temperatura all'uscita dall'intercooler (che può essere confusa con quella nel collettore di aspirazione).

Una nota sull'estrazione del consumo dai dati banco: con lunghe acquisizioni, a cui tutto l'algoritmo che sto presentando si riferisce, la bilancia che fornisce la misura di combustibile si svuota e viene quindi periodicamente riempita (operazione che avviene in automatico al raggiungimento di un certo livello). In questi frangenti, come si vede nella figura sottostante, viene mostrato un consumo negativo, che ovviamente deve essere "eliminato".

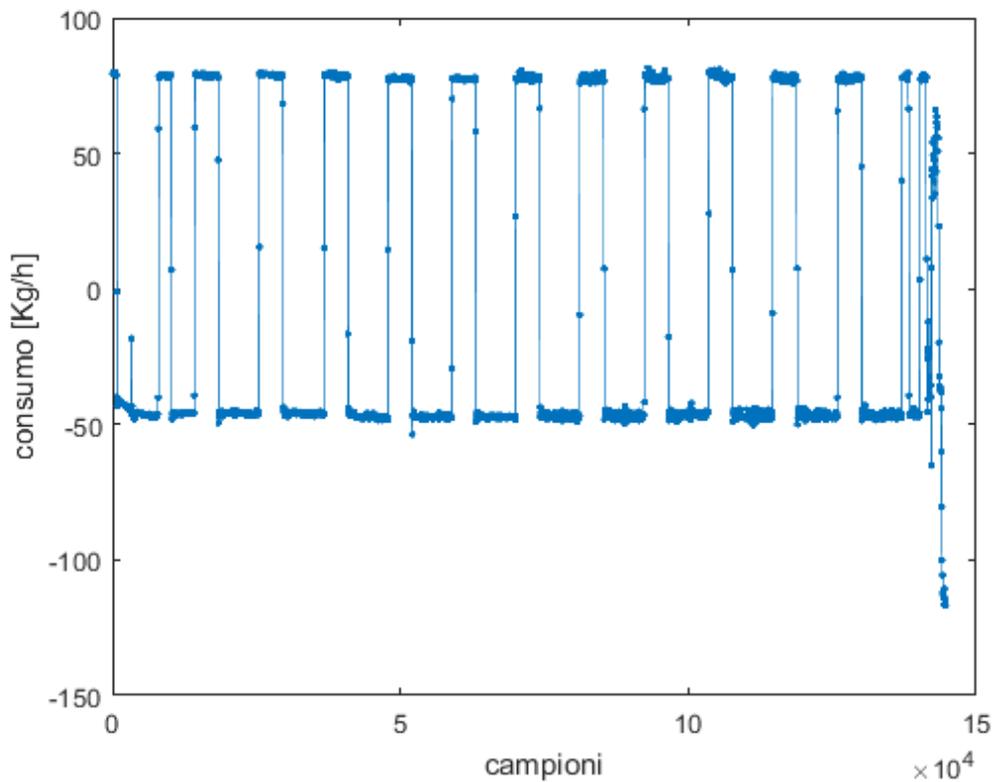


Figura 3.6

Il codice provvede quindi ad un trattamento del consumo prima della sua estrazione. In particolare, ai valori che corrispondono alle fasi di riempimento viene sommato un off-set per riportarli nella zona dei valori positivi, mantenendo comunque una certa informazione sul consumo. L'off-set è pari alla portata di riempimento della bilancia che, in condizioni stazionarie, può essere stimato pari alla differenza delle medie dei consumi in fase di riempimento (quando il consumo è negativo) ed in fase di svuotamento (cioè quando viene effettuata correttamente la misura). Viene inoltre operato un filtraggio passa-basso. La figura sottostante mostra il risultato del trattamento.

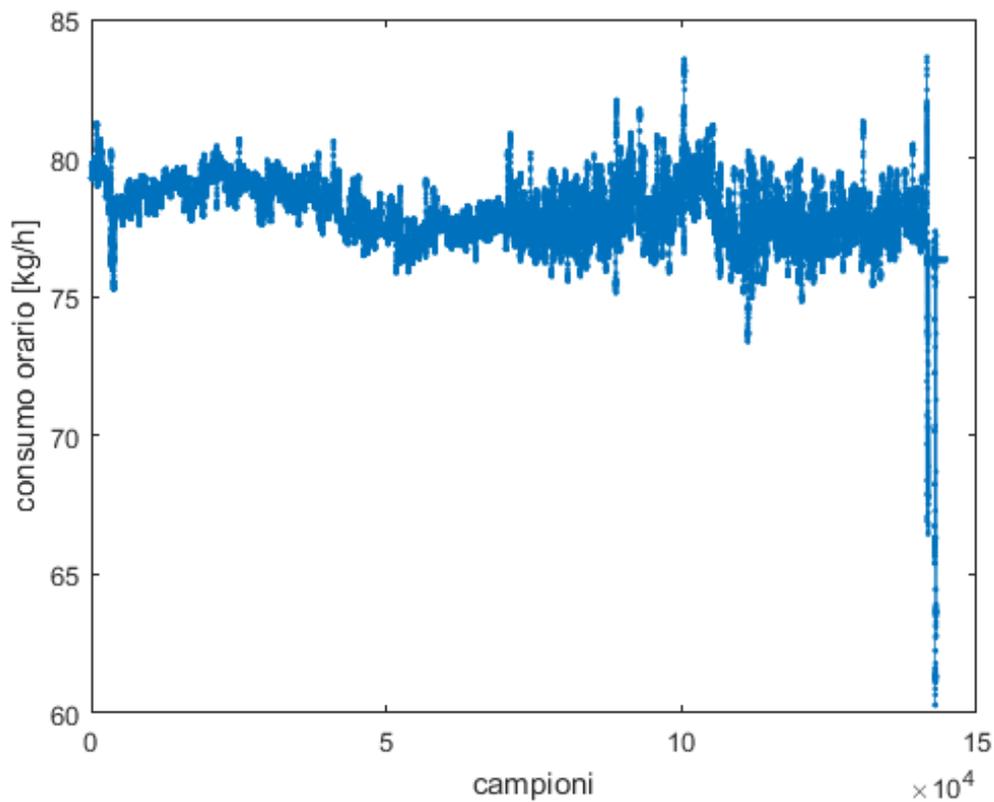


Figura 3.7

La figura 3.8 mostra un possibile risultato del codice di sincronizzazione. Sono rappresentati un dato banco (il consumo), un dato INCA (la portata d'aria) e un dato che deriva dall'acquisizione indicating (la PMI).

Come si vede, le acquisizioni INCA e banco partono prima e finiscono dopo rispetto all'acquisizione indicating, quindi hanno durata temporale diversa. Tuttavia, dopo aver effettuato la sincronizzazione, il codice realizza una corrispondenza univoca tra il generico ciclo motore e i dati INCA, banco e indicating.

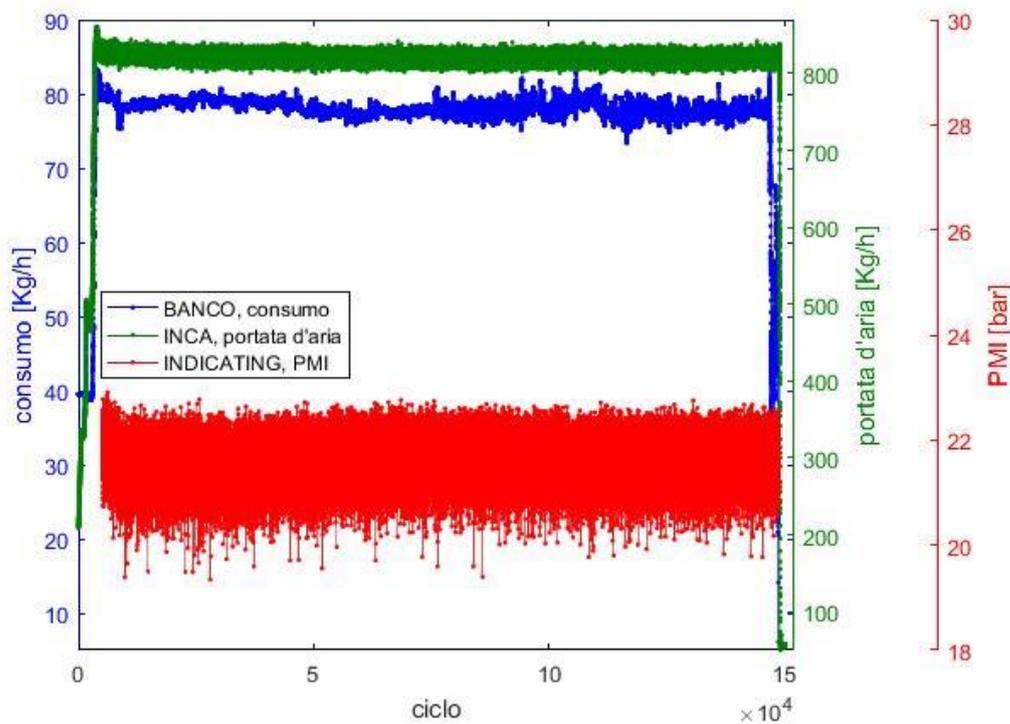


Figura 3.8

3.3 Interfaccia grafica per calcolo indici

Il risultato principale del mio lavoro di tesi è stato la realizzazione di un'interfaccia grafica in ambiente Matlab per il calcolo di indici relativi alla combustione (i CustomData).

L'obiettivo di questo lavoro è stato quello di realizzare uno strumento "semplice", cioè che possa essere utilizzato senza dover cambiare le righe di codice che effettuano i calcoli. L'idea fondamentale è quella che l'utente possa selezionare interattivamente gli indici che vuole calcolare, cambiare la configurazione ed effettuare altre operazioni senza dover modificare il codice.

Nello sviluppo dell'interfaccia ho perseguito anche l'obiettivo di ottenere una buona efficienza di calcolo, particolarmente importante se si considera la grande dimensione dei file indicating.

Vediamo allora come si presenta l'interfaccia al lancio.

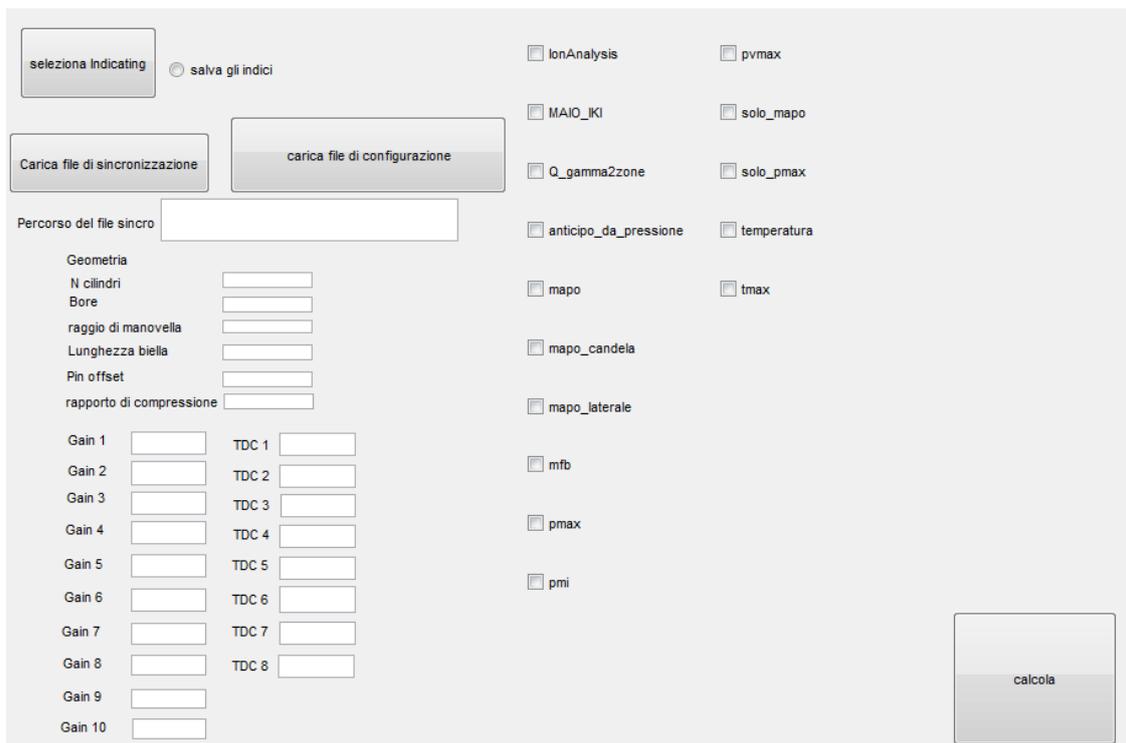


Figura 3.9

Ogni indice mostrato sull'interfaccia è associato ad una function Matlab.

Le function vengono caricate dinamicamente a partire dall'apposita cartella "funzioni". Ho realizzato delle function che calcolano indici "classici" (PMI, MAPO, Pmax...), e alcune specifiche per lo studio in questione (per esempio la function che analizza il segnale ION). Se l'utente vuole calcolare altri indici oltre a quelli già definiti, deve solo scrivere la function dedicata (seguendo una particolare sintassi di cui parlerò tra poco) e inserirla nella cartella "funzioni": il flusso di dati tra programma principale e function viene invece gestito autonomamente dall'interfaccia, come spiegherò in modo dettagliato.

In un'altra cartella, chiamata "supporto", sono presenti invece le function che servono appunto come supporto al calcolo di altre function, ma che non forniscono indici in uscita, e che quindi non sono visibili sull'interfaccia: tra queste, ci sono ad esempio la function che genera il riferimento angolare, quella che genera la funzione di volume, la funzione di filtraggio dei segnali...

Ad ogni indice sull'interfaccia è abbinata una checkbox, che deve essere spuntata se lo si vuole calcolare.

Premendo il tasto “seleziona indicating” si apre una finestra di dialogo che chiede di selezionare i file (o il file) mat indicating da elaborare.

Il bottone “salva gli indici” permette appunto di salvare in una cartella scelta dall’utente gli indici calcolati.

In relazione al salvataggio degli indici, l’algoritmo di funzionamento dell’interfaccia prevede 2 logiche di funzionamento diverse. La prima modalità prevede che il codice salvi il file contenente gli indici, per ogni file indicating, sovrascrivendo eventuali indici già calcolati.

Diversamente se invece si seleziona, attraverso la finestra di dialogo che si apre in automatico, di non voler sovrascrivere gli indici. In questa modalità di funzionamento, il codice controlla se esista o meno il file contenente gli indici relativi al file indicating corrente, prima di caricare il file indicating stesso. Nel caso il file con gli indici esista, verifica se ogni indice selezionato dall’utente sia già stata calcolato, e in tal caso il calcolo non viene ripetuto sul file indicating corrente. Al limite, se tutti gli indici sono già stati calcolati, il codice non carica il file indicating corrente, e passa ad effettuare il controllo sul file successivo.

Questa seconda modalità di salvataggio è stata sviluppata tenendo in considerazione il fatto che i dati da trattare possono essere di grande dimensione. I tempi di calcolo possono essere quindi particolarmente lunghi, e se per qualche motivo si interrompe l’elaborazione, può essere comodo non dover ricalcolare gli indici già calcolati.

L’interfaccia permette anche di selezionare il file di configurazione relativo ai dati che si vogliono analizzare.

Il suddetto file di configurazione viene generato da un apposito script (GeneraCH_CONFIG), che deve essere quindi modificato se si volessero cambiare dei parametri. I parametri sono in particolare quelli relativi alla geometria del motore (compresi i TDC), la corrispondenza tra ogni canale di acquisizione, i cilindri e i sensori, i relativi gain e la frequenza di campionamento. Nel caso dei dati a disposizione il setup di cella prevedeva l’utilizzo di una bancata sola di cilindri, quindi 4 cilindri. Su ogni cilindro erano poi installati 2 sensori di pressione (uno in candela, centrale, e uno sulla testa, laterale), per un totale di 8 canali relativi alla pressione. Gli ultimi due canali invece acquisivano i dati relativi alla corrente di ionizzazione (due cilindri su ogni canale).

Le tabelle sottostanti sono un esempio del contenuto del file di configurazione.

Canale	Cilindro	Sensore	TDC	Gain
1	1	Laterale	490	27,94
2	1	Centrale	490	95,51
3	2	Laterale	310	27,77
4	2	Centrale	310	96,34
5	3	Laterale	670	27,95
6	3	Centrale	670	96,25
7	4	Laterale	130	28,1
8	4	Centrale	130	96,4
9	1,3	ION		1
10	2,4	ION		1

Numero di cilindri	8
Alesaggio	86,5 mm
raggio di manovella	41 mm
lunghezza biella	143 mm
pin offset	0
rdc	9,4
frequenza di campionamento	200000 Hz

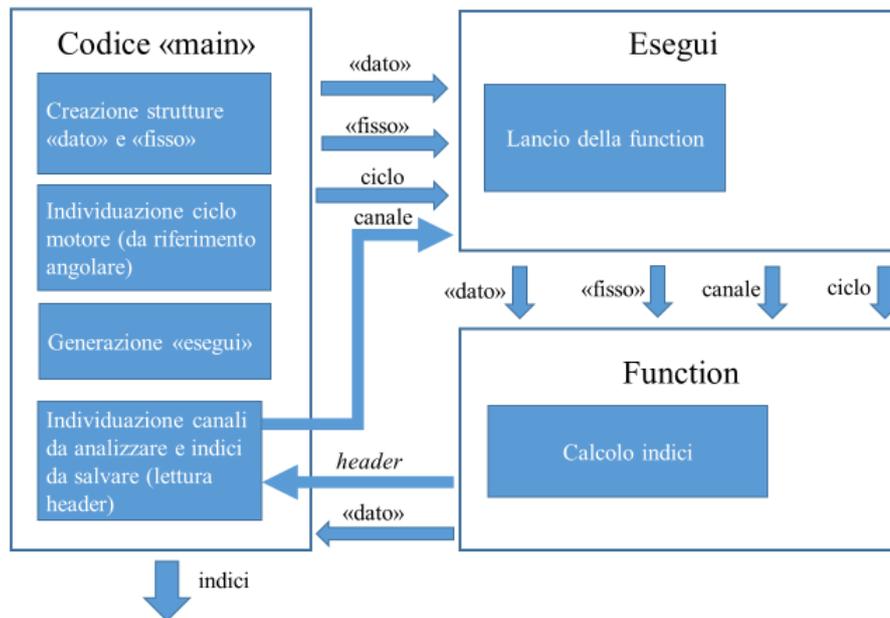
Il codice effettua un controllo di coerenza tra il contenuto del file di configurazione e i file indicating, confrontando i valori relativi alla geometria del motore, i *gain* e la frequenza di campionamento. In caso di non coerenza, viene mostrato un messaggio di errore: a questo punto si può scegliere di proseguire usando il file di configurazione, usando i parametri contenuti nei file indicating, o scegliere di non lanciare il calcolo. Nel caso si scelga di usare il file di configurazione, il contenuto di quest'ultimo viene mostrato sull'interfaccia.

È anche possibile modificare direttamente dall'interfaccia i parametri di configurazione: il successivo calcolo degli indici utilizzerà quindi i valori inseriti dall'utente. È importante sottolineare però che questa operazione non modifica il file di configurazione, ma ha influenza solo sul calcolo che si sta per lanciare.

È presente anche un pulsante che permette di selezionare il file di sincronizzazione (output dell'apposito codice che sincronizza i dati INCA, banco e indicating): il percorso del file selezionato viene poi mostrato nella *textbox* sottostante.

Entriamo ora più nel dettaglio del codice relativo dell'interfaccia.

Riporto uno schema che mostra come avvenga il flusso di dati tra il programma principale, le function e “esegui”, che definirò più avanti.



Il funzionamento del programma si basa sulla creazione di due strutture di dati Matlab, chiamate “dato” e “fisso”.

Come vagamente suggerisce il nome, nella struttura “fisso” vengono inseriti i valori che non variano per i ritagli di file indicating relativi alla medesima prova: per esempio, la configurazione dei canali e i dati geometrici del motore. La struttura “fisso” viene quindi generata solo una volta, quando vengono effettuati i calcoli relativi al primo file indicating selezionato. La struttura “fisso” è una delle variabili di input delle function.

La seconda struttura “dato”, invece, è realizzata *ad hoc* per ogni file indicating. In essa vengono inseriti appunto i dati necessari al lancio delle function che variano per ogni file indicating, e cioè ad esempio il segnale di pressione (filtrato e non), il riferimento angolare... Inoltre, ogni function inserisce i propri output all'interno della struttura “dato”.

La struttura “dato” è dunque sia un *input* che un *output* per ogni function. Ogni volta che una function viene chiamata, essa aggiorna la struttura “dato”, mettendo a

disposizione delle successive elaborazioni i risultati ottenuti, nonché trattamenti parziali sui segnali.

Per esempio, se viene lanciata la function che calcola il MAPO, essa provvede al filtraggio passa-alto del segnale di pressione, aggiornando la struttura “dato”: questa operazione viene però effettuata una volta sola (per file). Al successivo step di calcolo (per esempio per il ciclo successivo), il codice verifica se in “dato” sia già esistente la pressione filtrata passa-alto, e in caso esista non esegue nuovamente il filtraggio.

A questo punto si pongono però due problemi. Innanzitutto, ogni function deve conoscere su quali canali deve effettuare i calcoli.

In linea generale, infatti, avremo a disposizione (per questo specifico setup sperimentale) due segnali di pressione per ogni cilindro, uno relativo al sensore in candela e uno relativo al sensore laterale. Ebbene, alcune function effettuano i propri calcoli su uno solo dei due segnali (e quindi solo su alcuni canali), altre su entrambi (quindi su tutti i canali di pressione). C’è poi anche la function che analizza il segnale ION, che elabora solo i segnali relativi ai canali ION.

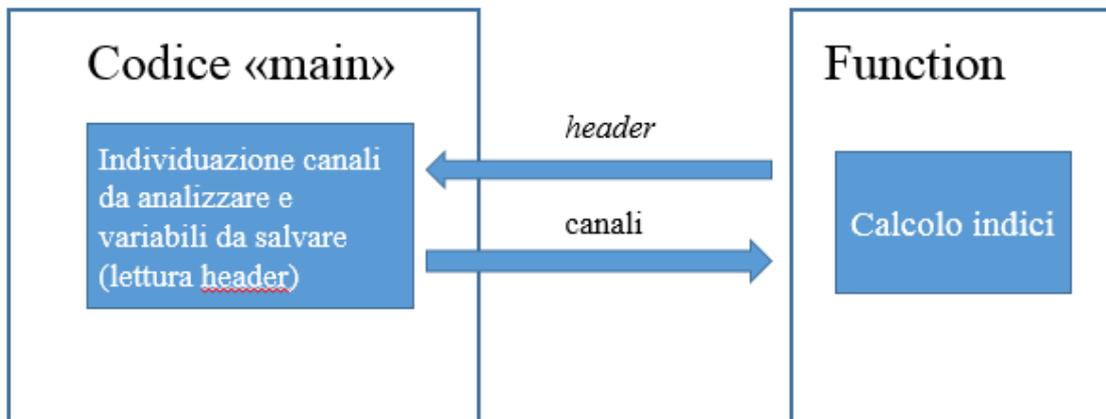
Ogni function inoltre, come ho detto, inserisce gli indici calcolati in “dato”: ma il codice ha bisogno di sapere quali siano i campi di “dato” che devono essere eventualmente salvati.

Per risolvere questi problemi si è pensato di inserire in testa ad ogni function due righe di *header*.

La prima riga indica su quali segnali (cioè su quali canali) bisogna effettuare i calcoli. Abbiamo deciso di indicare convenzionalmente con PP l’utilizzo di tutti i canali di pressione, con PC l’utilizzo dei solo canali di pressione prelevati in candela, con PL l’utilizzo dei soli canali di pressione prelevati dai sensori laterali (in testa), e con II l’utilizzo dei canali ION.

La seconda riga indica invece quali variabili verranno fornite in uscita, e permette quindi al codice di sapere quali saranno i campi della struttura “dato” che eventualmente dovrà salvare.

Il codice provvede dunque a “leggere” l’header di ogni funzione. In base a quello che legge “capisce” quali siano le variabili da salvare e fornisce in input ad ogni function solo i canali che devono essere analizzati.



Prima di lanciare le function il codice effettua delle operazioni “preliminari”, utilizzando anche alcune delle funzioni presenti nella già citata cartella di “supporto”. In ordine, si converte il segnale di pressione grezzo da volt a bar, attraverso un *gain* specifico per ogni canale.

Viene lanciata poi la funzione che genera il riferimento angolare per il file corrente (da cui discende anche il numero di cicli motore).

Vengono anche effettuati un filtraggio passa-basso del segnale di pressione (attraverso una apposita function) e il recupero della componente media dello stesso.

Se l’utente decide di usare anche le function che richiedono l’utilizzo dei dati banco e INCA, deve caricare il file di sincronizzazione prima di lanciare il calcolo, altrimenti il codice darà errore. In questo caso, a monte del lancio delle function il codice individua anche quali siano le porzioni di acquisizioni INCA e banco corrispondenti al file indicating che sta analizzando in quel momento. Questo viene realizzato mediante un controllo sui riferimenti ciclo, che sono contenuti nel file di sincronizzazione stesso. Anche i ritagli dei file INCA e banco vengono poi inseriti nella struttura “dato”.

Il codice a questo punto sa quali indici debbano essere calcolati (dalla selezione dell’utente ed eventualmente dal confronto con gli indici già esistenti), per quanti cicli motore (calcolati con la funzione di supporto Riferimento_angolare) e su quali canali (dalla lettura dell’header). Ogni function avrà dunque in ingresso 4 variabili: la struttura “fisso”, la struttura “dato”, il canale e il ciclo.

L’algoritmo che implementa il lancio delle function ha una struttura un po’ “intricata”, che trova giustificazione nell’efficienza del calcolo, come spiegherò più avanti.

In particolare, per ogni function da utilizzare il codice crea automaticamente, per mezzo di comandi *fprintf*, un altro codice *ad hoc*, chiamato “esegui”.

```
fid = fopen('esegui.m','wt');
fprintf(fid, 'function ');
fprintf(fid, '[dato] ');
fprintf(fid, ' = esegui(fisso,dato,canale,ciclo)\n');
fprintf(fid, '[dato] ');
fprintf(fid, ' = ');
fprintf(fid, funz(nn).name(1:end-2));
fprintf(fid, '(fisso,dato,canale,ciclo);');
```

“Esegui” ha come output solo la struttura “dato”, e come input le strutture “dato” e “fisso”, il canale e il ciclo. Il codice “esegui” effettua solo il richiamo alla function che deve essere utilizzata. Nel codice “esegui”, dunque, cambia di volta in volta solo il nome della function.

Riporto un esempio di codice “esegui”, creato per lanciare la function che calcola la PMI.

```
function [dato] = esegui(fisso,dato,canale,ciclo)
[dato] = pmi(fisso,dato,canale,ciclo);
```

Ogni indice selezionato viene quindi calcolato per ogni ciclo e per ogni canale.

Una volta terminati i calcoli relativi ad un certo indice, “esegui” viene eliminato automaticamente.

Ogni function viene chiamata con un comando del tipo

```
dato = esegui (fisso, dato, canale, ciclo)
```

da cui si vede appunto che le variabili di output vengono inserite nella struttura “dato”.

Ovviamente questo comporta che le function debbano essere scritte in modo appropriato, cioè le variabili di uscita debbano essere indicate come campi di “dato”.

Ma spieghiamo perché l’interfaccia lanci le function per il calcolo degli indici in questo modo.

Inizialmente il codice era impostato in modo diverso, e forse più “lineare”. La struttura del codice principale era la stessa, ma il modo in cui venivano lanciate le function era diverso.

Avevo infatti pensato di usare un comando `eval`, che esegue delle espressioni Matlab scritte sotto forma di stringhe di testo, ed è quindi molto versatile, ma è anche tipicamente “lento”.

La function sarebbe stata lanciata così

```
eval ([dato = nome_della_funzione(fisso,dato, canale, ciclo)])
```

Una struttura del genere non necessita di scrivere per ogni function il codice “esegui”. Tuttavia, effettuando delle prove ho notato che questa struttura impiega circa 1 secondo in più rispetto alla struttura con “esegui”, che non richiede l’utilizzo di comandi *eval*.

Considerato che questa interfaccia verrà usata per analizzare dati di dimensione considerevole, un risparmio dell’ordine di un secondo per ogni function può essere rilevante.

Una volta terminati i calcoli per la singola function, il codice salva (se così aveva deciso l’utente) gli indici calcolati. Vengono poi effettuate tutte le operazioni che ho descritto per le altre function eventualmente selezionate.

Vediamo ora per esempio come è stata impostata la function che provvede al calcolo del MAPO e di temperatura, angolo e pressione di knock onset.

```
%PP
%MAPO, AKO, PKO, TKO

function [dato] = mapo(fisso,dato,canale,ciclo)

k_start_P=0;
k_dur_P=70;

WS =find(dato.angolo_can(canale,dato.rifcicli_T_can(canale,ciclo):...
dato.rifcicli_T_can(canale,ciclo+1)-1) > k_start_P,1, 'first')+ ...
dato.rifcicli_T_can(canale,ciclo)-1;

WD = round(k_dur_P/dato.DTeta);
```

In questo caso, la prima riga di header indica che il calcolo va effettuato su tutti i canali di pressione, sia quelli relativi al sensore in candela, sia quelli relativi al sensore laterale.

La seconda riga invece indica le quattro variabili di uscita della function.

```

if not(isfield(dato, 'PP_004000_023000_04_filt_butter'))

    dato.PP_004000_023000_04_filt_butter = ...
    FUN_FILTER(dato.Time_Domain_T_G(1:numel(fisso.canali_P(:,1)),:),'...',...
    [4000 23000],4, fisso.freq_c, 'filt');

end

```

Da notare il controllo sull'esistenza o meno di una variabile all'interno della struttura "dato". In questo caso si tratta del segnale di pressione filtrato (attraverso apposita function di supporto FUN_FILTER). Il richiamo alla funzione di filtraggio avverrà dunque una volta sola (per file).

```

dato.MAPO(ciclo, canale) = ...
max(abs(dato.PP_004000_023000_04_filt_butter(WS:WS+WD, canale)));

[~, doveKO] = max(abs(dato.PP_004000_023000_04_filt_butter(WS:...
WS+WD, canale)));

dato.AKO(ciclo, canale) = dato.angolo_can(canale, WS+doveKO-1);

dato.PKO(ciclo, canale) = ...
dato.PP_000000_002000_04_filtfilt_butter_Recuperato(WS+...
doveKO-1, canale);

if not(isfield(dato, 'temperatura'))
    dato = FUN_temperatura(dato);
end

dato.TKO(ciclo, canale) = dato.temperatura(WS+doveKO-1, canale);

```

Lo stesso discorso vale per l'andamento della temperatura (calcolata anch'essa con una function dedicata), che bisogna conoscere per il calcolo di TKO.

Si noti infine che le variabili di uscita, in accordo con quanto ho detto prima, vengono fornite in output come campi della struttura dato.

4 Il segnale di pressione

Il segnale di pressione in camera di combustione è in assoluto il più ricco di informazioni, ed è l'unico che indica l'avanzamento della combustione.

Dal trattamento del segnale di pressione si possono ricavare molti indici che descrivono il funzionamento del motore. Tra questi, la coppia indicata (e dunque la PMI), indici che descrivono la durata della combustione, indici relativi alla detonazione, l'andamento della temperatura...

4.1 Sensori e catena di misura

Abbiamo detto che il segnale di pressione è ricco di informazioni, ma esso è di norma disponibile solo in sala prove, e non in vettura. Infatti i sensori affacciati in camera di combustione hanno un costo elevato e una durata limitata, tali da non rendere economicamente giustificabile la loro installazione in vettura (si usano solo in fase di sviluppo).

I sensori di pressione sono elementi molto delicati, che devono sopportare gradienti enormi di temperatura e pressione (centinaia di bar e migliaia di gradi in tempi dell'ordine del millisecondo).

Possono essere affacciati in camera di combustione direttamente forando la testata del motore, oppure possono essere inseriti in apposite candele strumentate. In linea generale, i sensori in candela presentano prestazioni peggiori.

Esistono due tipi fondamentali di sensori di pressione. I sensori raffreddati, come suggerisce il nome, presentano un apposito circuito di raffreddamento, che li rende più affidabili e resistenti. Di contro, questi sensori sono ingombranti, e possono presentare problemi di installazione.

I sensori non raffreddati, invece, sono meno resistenti, ma meno ingombranti, e quindi possono essere inseriti nelle apposite candele strumentate (utilizzabili eventualmente anche in vettura in fase di sviluppo).



Figura 4.1.



Figura 4.2. Candela strumentata

I sensori di pressione sono costruiti con materiale piezoelettrico (generalmente quarzo), che ha la caratteristica di generare una differenza di potenziale se viene sottoposto a deformazione.

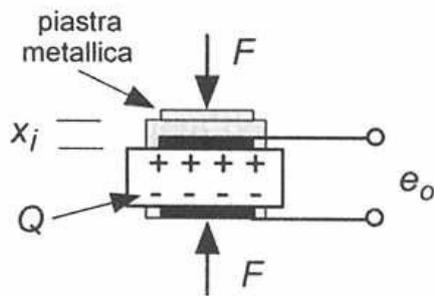


Figura 4.3

La quantità di carica Q che si accumula sulle facce dell'elemento piezoelettrico è proporzionale allo spostamento x , e dunque alla pressione P :

$$Q = k_x \cdot x = k_p \cdot P \quad (4.1)$$

Se si deriva rispetto al tempo l'equazione 3.1 si ottiene l'espressione

$$\frac{dQ}{dt} = i(t) = k_p \cdot \frac{dP(t)}{dt} \quad (4.2)$$

che mostra che se c'è una variazione di pressione si genera corrente.

Il sensore di pressione, dal punto di vista elettrico, è quindi assimilabile ad un generatore di corrente.

Il segnale in uscita è generalmente molto debole (ordine di grandezza 10 pC/bar), e deve essere quindi opportunamente amplificato.

A tale scopo si usa un amplificatore di carica, che permette di ottenere un segnale più forte, introducendo però delle distorsioni nello stesso.

L'amplificatore di carica è un amplificatore operazionale collegato in retroazione con un parallelo RC.

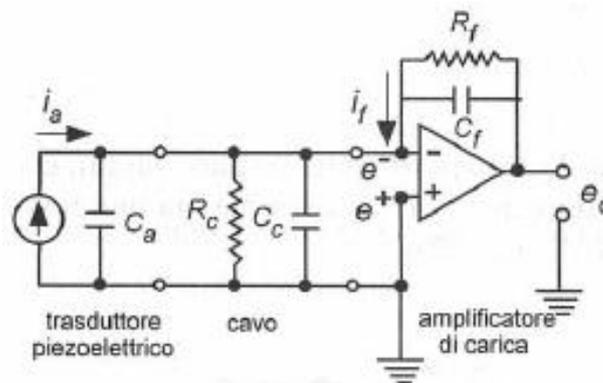


Figura 4.4

Con riferimento alla figura 4.4, il sensore è rappresentato dal generatore di corrente, C_a è la capacità del sensore stesso, mentre R_c e C_c sono rispettivamente la resistenza e la capacità del cavo.

Il collegamento in retroazione genera una condizione di “cortocircuito virtuale”, tale per cui si può considerare che i punti e^- ed e^+ siano allo stesso potenziale, e quindi le correnti che scorrono in C_a , R_c e C_c sono nulle. Possiamo così considerare uno schema semplificato equivalente.

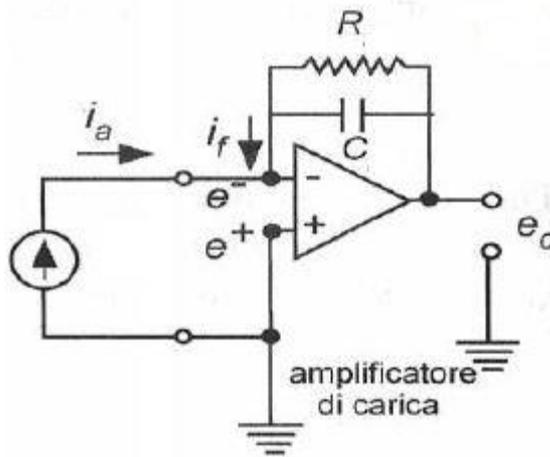


Figura 4.5

Se ora applichiamo la legge di Kirchoff al nodo e⁻

$$i_A = K_p \cdot \frac{dP(t)}{dt} = i_C + i_R = \frac{dQ_C}{dt} + \frac{e_0}{R} = C \cdot \frac{de_0}{dt} + \frac{e_0}{R} \quad (4.3)$$

da cui, esplicitando la derivata della pressione

$$\frac{dP(t)}{dt} = \frac{C}{K_p} \cdot \frac{de_0}{dt} + \frac{e_0}{R \cdot K_p} \quad (4.4)$$

Se ora integriamo l'espressione 4.4 tra il tempo 0 e il generico tempo t possiamo scrivere

$$P(t) = \frac{C}{K_p} \cdot e_0(t) + \frac{1}{R K_p} \cdot \int_0^t e_0(\tau) d\tau + P(0) \quad (4.5)$$

Analizziamo i termini a secondo membro. Il primo termine è legato al fattore di amplificazione della catena di misura (il guadagno), che è un valore noto in base ai settaggi dell'amplificatore di carica.

Il secondo termine è una distorsione del segnale, che interessa le bassissime frequenze.

Il termine P(0) è invece il termine a frequenza nulla, cioè il valore medio del segnale di pressione.

Dato che il segnale di pressione è fondamentalmente periodico, può essere espresso come somma di seni e coseni, applicando cioè la trasformazione secondo Fourier.

$$P(j\omega) = \frac{C}{K_p} E_0(j\omega) + \frac{E_0(j\omega)}{R K_p j\omega} = \frac{C}{K_p} \frac{(1 + RCj\omega)}{RCj\omega} E_0(j\omega) \quad (4.6)$$

Da cui si ottiene la funzione di trasferimento

$$\frac{E_0(j\omega)}{P(j\omega)} = G(j\omega) = \frac{K_p}{C} \cdot \frac{RCj\omega}{(RCj\omega + 1)} \quad (4.7)$$

che è la funzione di trasferimento tipica di un filtro passa-alto, con costante di tempo RC e frequenza di taglio $\frac{1}{2\pi RC}$.

L'amplificatore di carica si comporta dunque come un filtro passa-alto, tende cioè a distorcere e deamplificare le basse frequenze. La realizzazione tecnologica consente però di avere distorsione solo per frequenze bassissime, e quindi in generale questo non è un grosso problema. Rimane però il fatto che la componente media del segnale di pressione, a frequenza nulla, non passa, e quindi in uscita si ha un segnale di pressione a valor medio nullo (cioè anche con pressioni negative), che non ha senso fisico. Per il calcolo di alcuni indici è necessario il recupero della componente media, di cui parlerò diffusamente più avanti.

4.2 Acquisizione in sala prove

Nella sala prove dell'Università di Bologna il segnale di pressione viene acquisito attraverso l'hardware ed il relativo software OBI (On Board Indicating).

L'attuale configurazione di cella prevede l'acquisizione dei dati di pressione relativi ad una sola bancata di cilindri, in particolare la bancata sinistra. In ogni cilindro sono installati due sensori di pressione, uno in candela e uno laterale (attraverso apposita foratura della testata del motore), per un totale di otto canali di pressione.

Il segnale viene acquisito alla frequenza di 200 kHz, quindi un campione ogni 5 microsecondi. Da una così alta frequenza di campionamento discende l'elevata dimensione dei file prodotti in uscita dal programma di acquisizione. Il file in uscita contiene il segnale di pressione "grezzo" in formato streaming, cioè con tutti i cicli uno di fila all'altro.

4.3 Trattamento del segnale di pressione

4.3.1 Recupero della componente media

Per il recupero della componente media ho usato nei codici il metodo della politropica ad esponente forzato. Si fa l'ipotesi che in fase di compressione, quando la combustione non è ancora iniziata (quindi non ci sono reazioni esotermiche), e con le valvole chiuse (quindi la massa è circa costante) valga una legge politropica del tipo

$$P \cdot V^n = \text{costante} \quad (4.8)$$

Se ora applichiamo l'equazione politropica a due punti generici 1 e 2 che soddisfano le ipotesi sopra fatte, possiamo scrivere

$$P_{1,vera} \cdot V_1^n = P_{2,vera} \cdot V_2^n \quad (4.9)$$

e cioè

$$(P_1 + \Delta P) \cdot V_1^n = (P_2 + \Delta P) \cdot V_2^n \quad (4.10)$$

Nell'equazione 4.10 l'unica incognita è ΔP , che è proprio la componente media del segnale di pressione.

Il valore di ΔP così ottenuto viene sommato a tutto il segnale di pressione.

Nei codici vengono usati come angoli di riferimento 100° e 50° BTDC. La figura 4.6 rappresenta un esempio del risultato dell'applicazione di questo metodo.

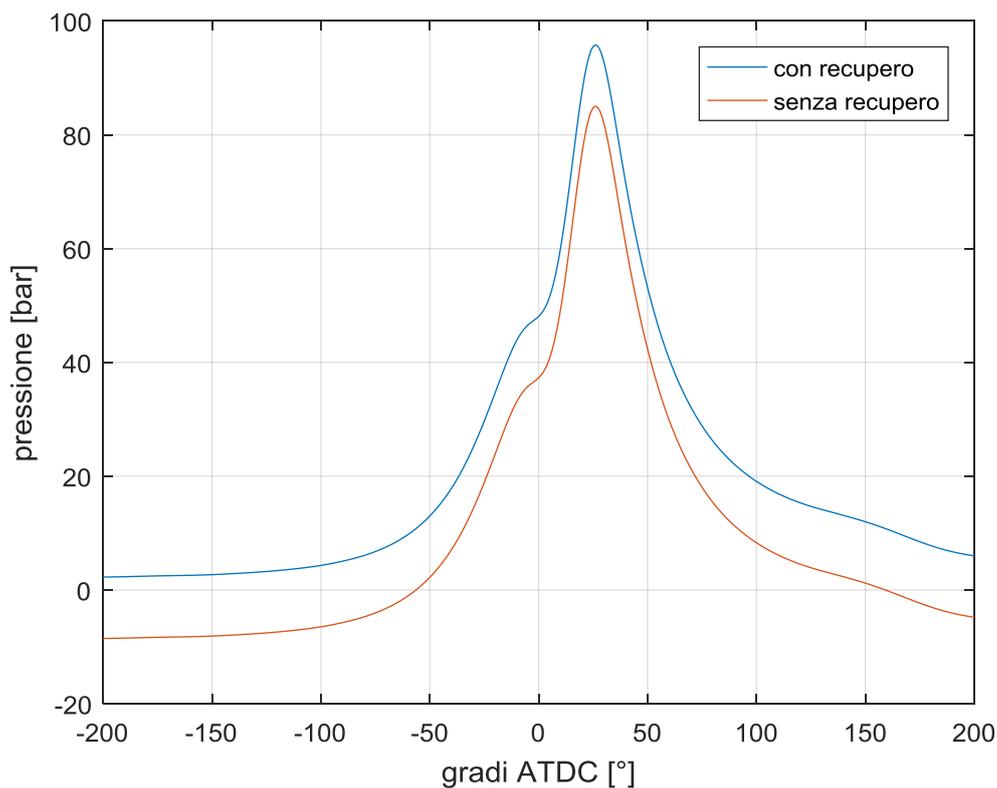


Figura 4.6

Per irrobustire il metodo si potrebbero considerare più valori di pressione e volume. Il limite principale rimane comunque la determinazione dell'esponente della politropica: un valore di riferimento per i motori ad accensione comandata è 1.34.

4.3.2 Conversione volt bar

Il segnale grezzo contiene l'informazione in volt, che viene trasformata in bar attraverso i guadagni (*Gain*), che in linea generale sono diversi per ogni canale.

4.3.3 Generazione del riferimento angolare

Il segnale di pressione grezzo presenta, come detto, tutti i cicli uno in fila all'altro, e non contiene nessun riferimento angolare. Per poter identificare i vari cicli occorre quindi generare a parte il riferimento angolare. Questa operazione viene realizzata nei codici dall'apposita function di supporto "Riferimento_angolare".

Il file *indicating* contiene anche l'informazione relativa ai tempi dente, grazie alla quale si possono trovare i riferimenti ciclo (e giro). Con i riferimenti ciclo si costruisce poi il vettore "ANGOLO", che rappresenta il riferimento angolare "assoluto": esso associa ad ogni campione di pressione l'angolo (assoluto) tra 0 e 720. Il vettore ANGOLO vale 0 in corrispondenza del secondo dente dopo il buco della ruota fonica (perché sulla fonica mancano 2 denti).

La ruota fonica ha un numero finito di denti, nel caso specifico (ma è uno standard in ambito automotive) 60 (di cui 2 "virtuali"). La risoluzione angolare è dunque di 6 gradi. Il posizionamento angolare dei campioni sarà "preciso" solo per i campioni che si trovano in prossimità (al limite corrispondenti) al passaggio di un dente della ruota fonica. Per gli altri denti si effettua una interpolazione lineare tra il dente precedente e il successivo, facendo l'ipotesi (sufficientemente veritiera) che la velocità di rotazione sia costante tra due denti.

La function "Riferimento_angolare" genera come detto il riferimento *assoluto*. Il riferimento *relativo* per ogni cilindro (che va da -360 a +360, con 0 in corrispondenza del relativo PMS di fine compressione) viene poi calcolato nel codice principale, nota la posizione dei TDC cilindrici e dell'eventuale ATDC. L'ATDC è l'angolo tra la manovella e l'asse del pistone quando esso si trova al punto morto superiore. L'ATDC è quindi calcolato come l'arco il cui seno è il rapporto tra il *pin-offset*, cioè la distanza tra il centro di rotazione della manovella e l'asse del pistone, e la somma delle lunghezze di biella e manovella.

La motivazione che spinge a costruire cinematismi biella-manovella con *pin-offset* diverso da zero risiede nella possibile diminuzione degli attriti. L'entità dello

sfasamento tra centro di rotazione della manovella e asse del pistone è dell'ordine dell'1% del diametro del pistone.

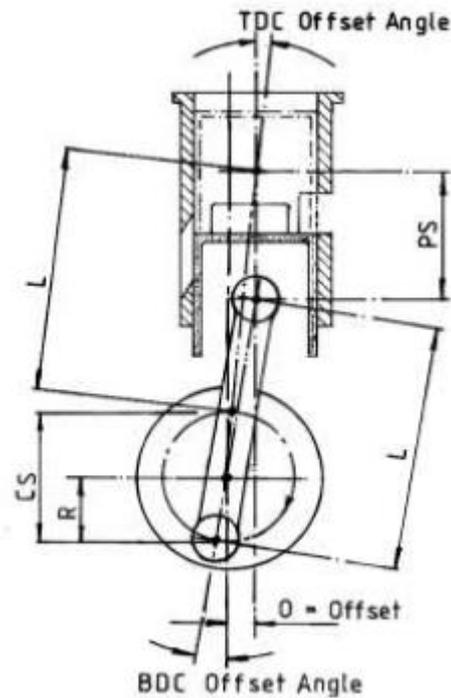


Figura 4.7

4.4 Utilizzo del segnale di pressione per il calcolo di indici

4.4.1 PMI

La function che calcola la PMI fa riferimento al segnale di pressione filtrato passa-basso.

Si calcola prima la coppia indicata, eguagliando il lavoro fatto dai gas sul pistone con quello fatto dalla coppia all'albero:

$$PdV = C_{ind}d\vartheta \quad (4.11)$$

Il calcolo della legge di volume viene effettuato da una apposita function di supporto, sulla base di relazioni geometriche derivanti dall'analisi del cinematismo biella-manovella.

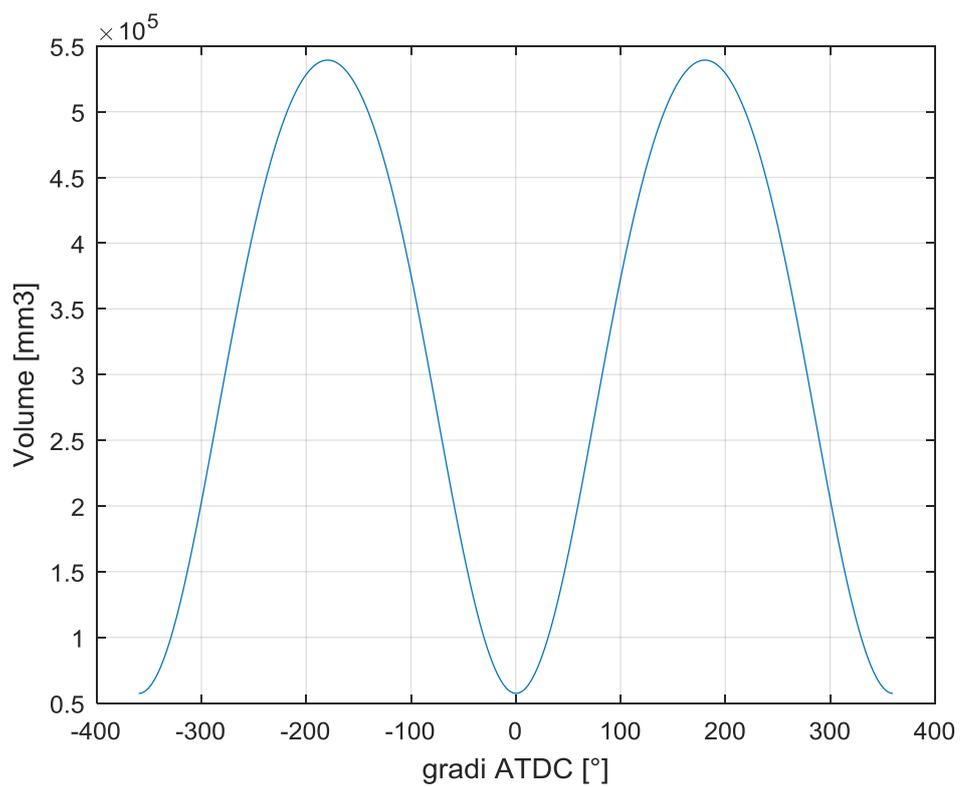


Figura 4.8. Legge di volume.

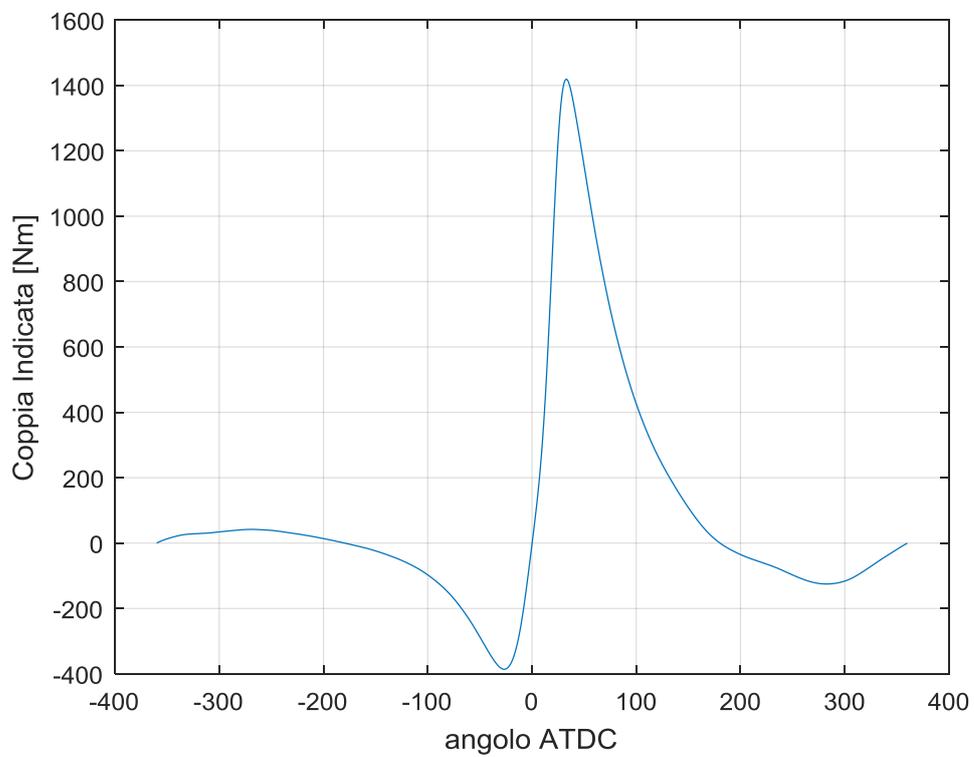


Figura 4.9. Coppia indicata.

La PMI si calcola poi dividendo il lavoro totale del ciclo (cioè la cumulata della coppia indicata sull'arco angolare 4π) per la cilindrata unitaria.

4.4.2 Pmax

È la pressione massima raggiunta in camera, e si ottiene dal segnale di pressione filtrato passa-basso. Dà indicazione sul livello di pressione raggiunto in camera, a prescindere dalla bontà della conversione in lavoro, informazione invece contenuta nella PMI.

4.4.3 Temperatura

Applicando l'equazione di stato dei gas perfetti (e considerandoli ideali) è possibile stimare il profilo medio di temperatura in camera di combustione ed eventualmente estrarne il valore massimo. Questo calcolo richiede la conoscenza della massa nel cilindro, calcolabile a partire dalle informazioni sulla massa di combustibile e sul titolo (quindi dalle acquisizioni INCA e banco).

L'accuratezza di questo calcolo può essere influenzata da valori non uniformi del consumo e del titolo tra i vari cilindri, oltre che dalla validità del modello usato per la stima di R (costante del gas).

4.4.4 Indici di detonazione

Il segnale di pressione è un indicatore diretto della presenza del fenomeno della detonazione. Cicli detonanti presentano infatti una caratteristica oscillazione ad alta frequenza della pressione.

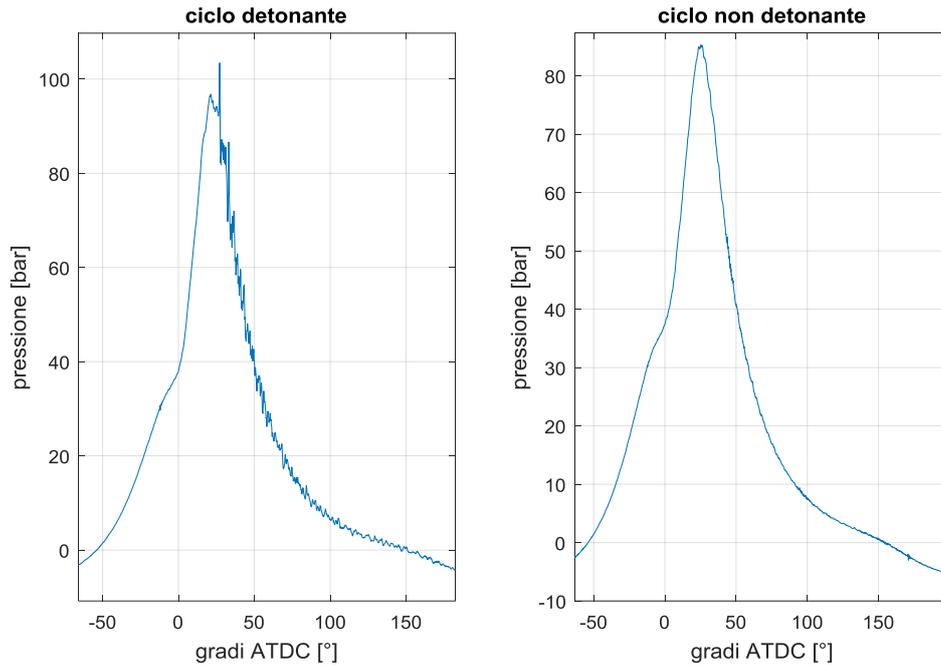


Figura 4.10

Il MAPO (*Maximum Amplitude of Pressure Oscillation*) è l'indice di detonazione più comune, ed è definito come la massima ampiezza di oscillazione della componente ad alta frequenza del segnale di pressione.

$$\text{MAPO} = \max (\text{abs} (P_{\text{hf}})) \quad (4.12)$$

Le function che calcolano il MAPO operano un filtraggio passa-banda del segnale di pressione (le frequenze di filtraggio dipendono da quali modi di vibrare si vogliono evidenziare). Il segnale viene poi finestrato tra 0 e 70 °ATDC, e in questo intervallo si prende il massimo del valore assoluto.

Una delle function effettua il calcolo su tutti i canali di pressione. In questo caso viene utilizzata la stessa banda passante sia per i segnali di pressione provenienti dai sensori in candela che per quelli provenienti dai sensori laterali.

La function in questione fornisce in uscita anche la temperatura, l'angolo e la pressione corrispondenti all'istante di Knock onset. Con il termine *Knock onset* si indica l'inizio (*onset*) della detonazione: non potendo tuttavia localizzare con precisione l'inizio dell'evento detonante, si confonde questo istante con quello corrispondente al MAPO.

Ci sono poi altre due function che sono invece dedicate al calcolo del MAPO per uno solo dei due segnali di pressione. Una elabora il segnale proveniente dal sensore in candela, l'altra quello proveniente dal sensore laterale.

In questo caso le due function usano bande di frequenze diverse per il filtraggio, perché i sensori in candela e laterali rilevano diversi modi di vibrare della camera. Ad esempio, il sensore in candela non rileva (o rileva in modo molto attenuato) tutti i modi di vibrare che presentano un nodo al centro della camera.

Nella tabella sottostante sono riportate alcune frequenze proprie della camera di combustione del motore in oggetto.

Frequenze rilevate [kHz]	
Testa	Candela
6,9	6,9
11,6	
14,1	13,9
15,5	
19,2	19,2
23,6	23,6

Queste function effettuano un filtraggio del segnale su bande di frequenza di ampiezza 1 kHz, centrate sui valori di eccitazione dei modi di vibrare.

In uscita si avranno tanti indici quante sono le bande di frequenza considerate.

A valle dell'utilizzo di queste due function si può quindi definire un indice SDBP (*Sum of Divided Band-Pass*), che è la somma della massima ampiezza di oscillazione in ognuna delle bande (cioè la somma dei MAPO).

4.4.5 Legge di rilascio calore

Si può dimostrare che sempre a partire dal segnale di pressione si possono ottenere informazioni termodinamiche, in particolare sul rilascio di calore.

L'analisi parte da un bilancio energetico in camera di combustione (ipotizzando che le valvole siano chiuse). Se facciamo riferimento alla figura 4.11, possiamo scrivere

$$\delta Q_{comb} = dU + \delta Q_{wall} + dL + hdm \quad (4.13)$$

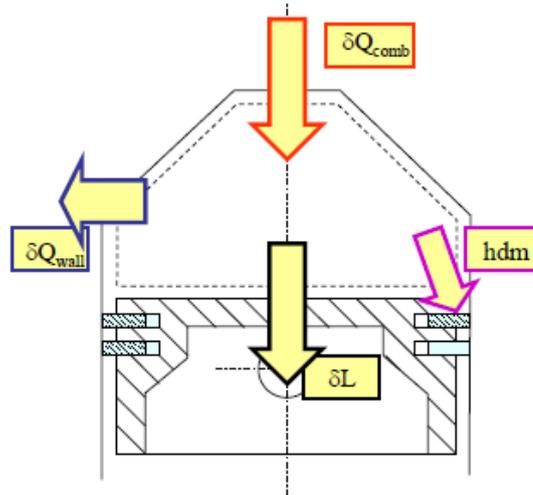


Figura 4.11

L'equazione 4.13 significa che il calore *lordo* rilasciato dalla combustione è uguale alla somma di 4 termini: in ordine, la variazione di energia interna del gas, il calore disperso alle pareti, il termine “utile” dL (cioè il lavoro effettuato sul pistone), e la variazione di entalpia dovuta ai trafilamenti sulle fasce. Esplicitando l'energia interna e il lavoro

$$dU = d(m \cdot u(T)) = mc_v dT - u(T) \cdot dm \quad (4.14)$$

e

$$dL = pdV \quad (4.15)$$

Possiamo quindi scrivere

$$\delta Q_{comb} = mc_v dT + pdV + \delta Q_{wall} + (h - u)dm \quad (4.16)$$

In quest'ultima espressione, il calore scambiato a parete e il termine $(h-u)dm$ rappresentano delle “perdite”. Se definiamo quindi il calore “netto” con la seguente espressione

$$\delta Q_n = \delta Q_{comb} - \delta Q_{wall} - (h - u)dm \quad (4.17)$$

possiamo anche scrivere che esso è

$$\delta Q_n = mc_v dT + pdV \quad (4.18)$$

Ora, se applichiamo l'equazione di stato dei gas perfetti (considerando costante la massa nel cilindro) possiamo differenziare la temperatura e scrivere quindi

$$mc_v dT = \frac{mc_v}{mR} * (PdV + VdP) \quad (4.19)$$

Ricordando poi che

$$R = c_p - c_v$$

e

$$\gamma = c_p / c_v$$

possiamo giungere, dopo alcune manipolazioni algebriche, alla forma

$$\delta Q_n = \frac{\gamma}{\gamma-1} pdV + \frac{1}{\gamma-1} Vdp \quad (4.20)$$

L'espressione ricavata mostra dunque che per calcolare il rilascio di calore "netto" bisogna usare la pressione, il volume e γ .

Se consideriamo l'espressione 4.20 per il generico angolo infinitesimo $d\theta$, otteniamo l'espressione del ROHR (*Rate Of Heat Release*), che esprime la velocità (espressa in $J/^\circ CA$) con cui viene introdotto calore nel sistema attraverso il processo di ossidazione del combustibile.

$$ROHR = \frac{\gamma}{\gamma-1} P \frac{dV}{d\theta} + \frac{1}{\gamma-1} V \frac{dP}{d\theta} \quad (4.21)$$

La figura 4.12 mostra un esempio di andamento del ROHR (l'angolo 0 è il PMS).

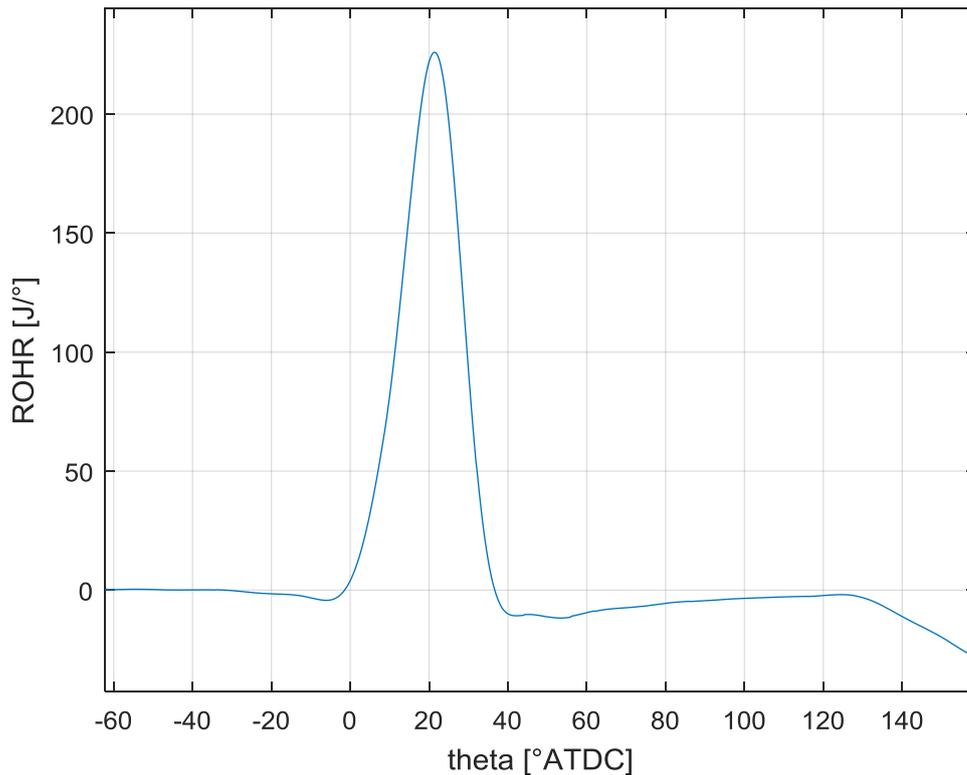


Figura 4.12

In questo caso si vede che da circa 40° dopo il punto morto superiore inizia una zona in cui il rilascio di calore è negativo. Se integriamo il ROHR dal punto in cui diventa negativo fino a circa 120° (a 130° circa si aprono le valvole di scarico, e quindi non vale più l'ipotesi di massa costante) possiamo ricavare una stima del calore scambiato a parete.

È bene precisare però che un calcolo di questo tipo considera solo gli scambi a parete che avvengono *dopo* la fine della combustione: in generale, invece, ci saranno perdite a parete anche *durante* la combustione.

Se si effettua poi la cumulata del ROHR si ottiene l'andamento del calore totale rilasciato (netto), chiamato CHR (*Cumulative Heat Release*).

La function fornisce in uscita il valore massimo del CHR, che indica il calore netto introdotto in camera di combustione.

Per il calcolo degli indici termodinamici è fondamentale il modo con cui viene calcolato il valore di γ , cioè il rapporto tra i calori specifici a pressione e a volume

costante. L'accuratezza del bilancio di energia dipende infatti pesantemente da come vengono rappresentati i cambiamenti nell'energia interna della carica cilindrica. In letteratura sono stati proposti vari modelli per la determinazione dei calori specifici, con diverse ipotesi di base e che comportano una differente complicazione a livello computazionale.

La function in questione fa riferimento ad un modello "a 2 zone", in cui si distinguono la zona con i gas incombusti ("unburnt") e quella con i gas combusti ("burnt"), separate da uno strato infinitesimo che rappresenta il fronte di fiamma¹. All'interno di ogni zona sono omogenee la temperatura e le proprietà termodinamiche, mentre la pressione è uguale in entrambe le zone.

Il valore di γ viene calcolato fondamentalmente per mezzo di relazioni polinomiali, in cui entrano i valori di temperatura, titolo e pressione.

La function in questione calcola dunque anche l'andamento della temperatura. Per questo motivo è necessario conoscere il valore della massa nel cilindro, che può essere calcolato usando la massa di combustibile e il titolo. Questi ultimi due dati sono ottenibili dalle acquisizioni INCA e banco.

Dal CHR si può anche ricavare poi un altro valore di grande interesse, l'MFB (Mass Fraction Burnt), definito come il rapporto tra la funzione CHR e il suo valore massimo. Di fatto l'MFB si ottiene normalizzando la curva del CHR rispetto al suo valore massimo.

Per il calcolo dell'MFB si limita l'analisi del CHR ad un arco angolare in cui si suppone che sia compresa tutta la durata della combustione e in cui le valvole siano chiuse (massa nel cilindro costante). Nei codici l'arco angolare scelto va da 20 °BTDC a 135 °ATDC. Questi angoli vengono utilizzati anche per il trattamento preliminare del CHR, funzionale proprio alla definizione dell'MFB.

La funzione CHR può infatti presentare nella parte iniziale dei valori negativi, perché la combustione non è ancora iniziata e ci sono degli scambi a parete. L'analisi dell'MFB dovrebbe partire allora dal punto di minimo. Si porta quindi il punto di minimo a coincidere con lo zero, sommando un off-set a tutta la curva, trascurando i campioni precedenti al minimo. I campioni successivi al valore di massimo vengono invece posti uguali proprio al valore massimo.

¹ Klein, M., Eriksson, L., "A Specific Heat Ratio Model for Single-Zone Heat Release Models" (2004).

Dalla curva dell'MFB si possono ottenere poi gli angoli corrispondenti ad una certa frazione di massa bruciata, indici estremamente utili per l'analisi della combustione. Una function apposita calcola gli angoli (CA_XX_MFB) corrispondenti al 1%, 10%, 50%, 90% e 95% di massa bruciata, cercando il primo valore sulla curva di MFB maggiore (o uguale) alla percentuale desiderata.

Nella function in questione si possono modificare i valori di CA MFB che si desidera calcolare, scrivendoli direttamente nella seconda riga di header. La function infatti "legge" le percentuali desiderate, che vengono poi calcolate per mezzo di comandi *eval*.

Sulla curva di CHR si potrebbe effettuare anche un altro tipo di analisi per ottenere una stima del calore scambiato a parete. In effetti, se si considerasse la curva di CHR (calore netto) anche dopo il punto di massimo, si vedrebbe che essa tende a calare, in accordo con i valori di ROHR negativi di cui ho parlato sopra (usati proprio per calcolare lo scambio a parete).

Se ora noi conosciamo la massa di combustibile iniettata (dato ottenibile dall'acquisizione banco sul consumo) e il potere calorifico del combustibile, possiamo calcolare il calore lordo introdotto in camera di combustione.

Lo scambio a parete sarebbe allora la differenza tra calore lordo e calore netto.

Da notare che in questo tipo di analisi le perdite dovute all'efficienza di combustione e al *blow-by* vengono incluse nelle perdite a parete.

Inoltre, il valore calcolato di scambio a parete dipende dall'angolo per cui si decide di considerare il CHR. Effettuando l'analisi fino ad un istante prima dell'apertura delle valvole di scarico si considerano le perdite a parete che si realizzano sia *durante* che *dopo* la combustione.

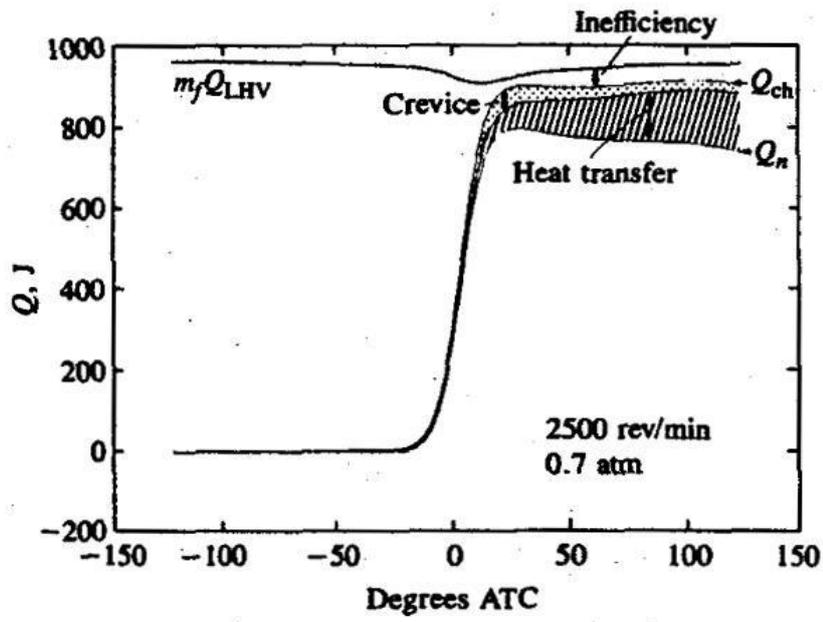


Figura 4.13

5 Il segnale ION

Abbiamo detto che il segnale di pressione è ricco di contenuto, e da esso si possono ottenere molte informazioni. Tuttavia in vettura non sono tipicamente presenti i sensori di pressione, per i già citati problemi legati al costo e all'affidabilità degli stessi.

Alcune delle informazioni ottenibili dal segnale di pressione, tuttavia, possono essere ottenute anche dal segnale di corrente di ionizzazione (ION), che invece è possibile avere in vettura.

L'utilizzo del segnale ION è una tecnologia relativamente recente (le prime applicazioni risalgono alla fine degli anni '80), e ancora oggi è utilizzata prevalentemente in vetture di alta gamma.

Il segnale ION rappresenta l'andamento della concentrazione di ioni liberi in camera di combustione.

Il sensore utilizzato per ottenere questo segnale è la candela stessa.

La misura avviene applicando una differenza di potenziale (di circa 500 V) agli elettrodi della candela e misurando la corrente che circola. In altre parole, si misura la conduttività della miscela, che varia in base alla ionizzazione (cioè la quantità di ioni liberi) della stessa.

La figura 5.1 rappresenta un esempio di andamento tipico del segnale ION.

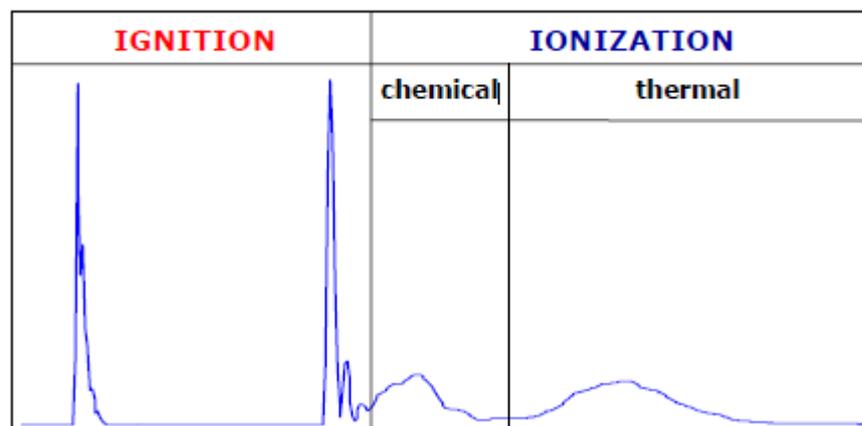


Figura 5.1

I primi due spike sono relativi alla fase di accensione, e in particolare corrispondono rispettivamente all'inizio della fase di carica della bobina e all'innesco della scintilla. Questa porzione di segnale in genere non viene analizzata, perché la combustione non è ancora iniziata.

La seconda parte, denominata “Ionization”, corrisponde invece agli angoli in cui avviene la combustione.

Sono presenti tipicamente due picchi, che si formano secondo diverse modalità:

- Meccanismo chimico, per cui la ionizzazione avviene in seguito a reazioni di ossido-riduzione;
- Meccanismo termico, in cui la dissociazione termica del monossido di azoto libera molti ioni.

Il picco della fase chimica indica che la combustione è partita, e corrisponde allo sviluppo delle reazioni di pre-fiamma (che liberano molti ioni).

Il secondo picco mostra invece che in camera di combustione c'è un'elevata ionizzazione come conseguenza delle alte temperature.

L'aspetto che rende interessante il segnale ION è che il picco termico è generalmente in fase con il picco della pressione. Quindi, limitando l'analisi del segnale alla parte “Ionization”, si trova in genere una elevata correlazione con il segnale di pressione.

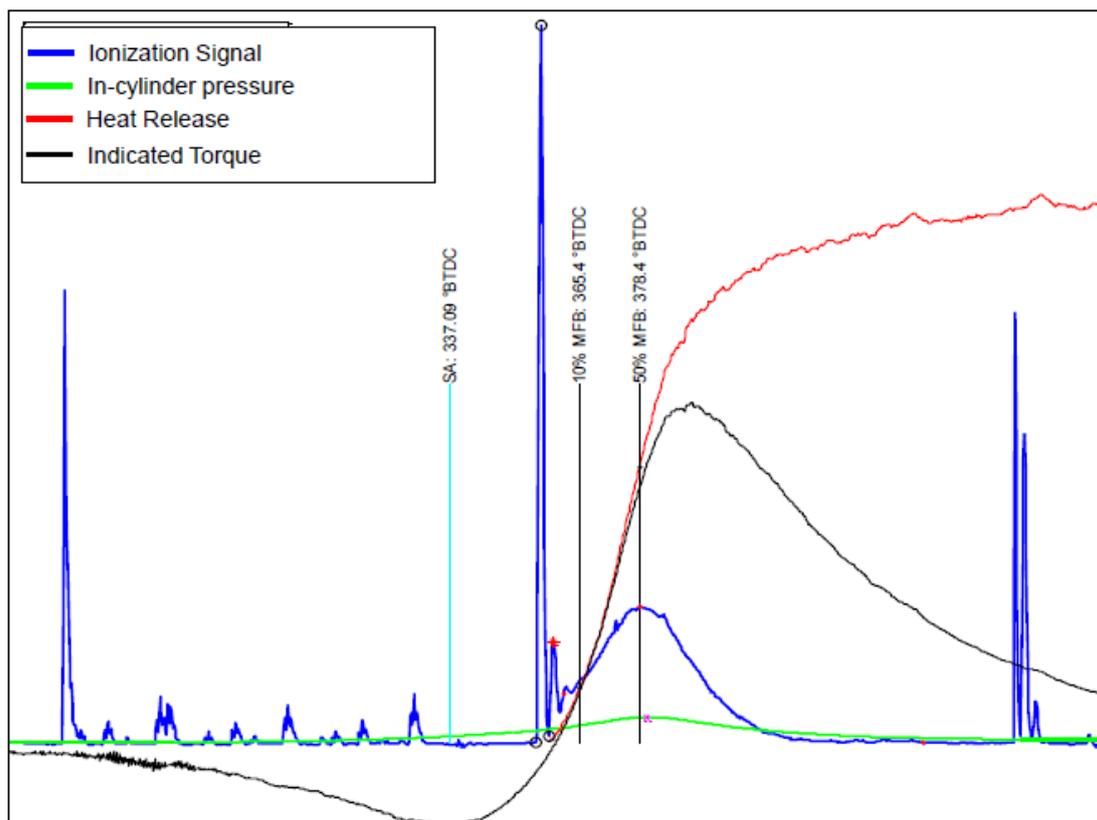


Figura 5.2

5.1 Trattamento del segnale ION

Nella sala prove dell'Università di Bologna il segnale ION è acquisito usando solo 2 canali per 4 cilindri. La porzione di segnale "interessante" è infatti contenuta in un arco angolare molto ristretto, come spiegherò meglio tra poco, quindi si possono acquisire segnali relativi a più cilindri usando un solo canale.

È possibile definire degli indici di detonazione basati sul segnale di corrente di ionizzazione, in modo simile a quanto viene fatto con il segnale di pressione.

Se prendiamo ad esempio in considerazione il MAPO, possiamo definire un indice "corrispondente" MAIO (*Maximum Amplitude of Ion Oscillation*), cioè il valore massimo del valore assoluto del segnale ION filtrato passa-alto.

Ovviamente, per definire il MAIO occorre restringere l'analisi all'arco angolare in cui avviene la detonazione, altrimenti verrebbero presi in considerazione gli spike di cui ho parlato precedentemente, o addirittura il segnale corrispondente all'altro cilindro acquisito sullo stesso canale. Una finestra plausibile va da 10° ATDC a 70° ATDC.

Il MAIO, in modo analogo al MAPO, può essere usato come indicazione dell'intensità dell'evento detonante.

Definire il MAIO ha senso perché le oscillazioni causate dalla detonazione sul segnale di pressione, sono visibili anche sul segnale ION, come mostra la figura sottostante.

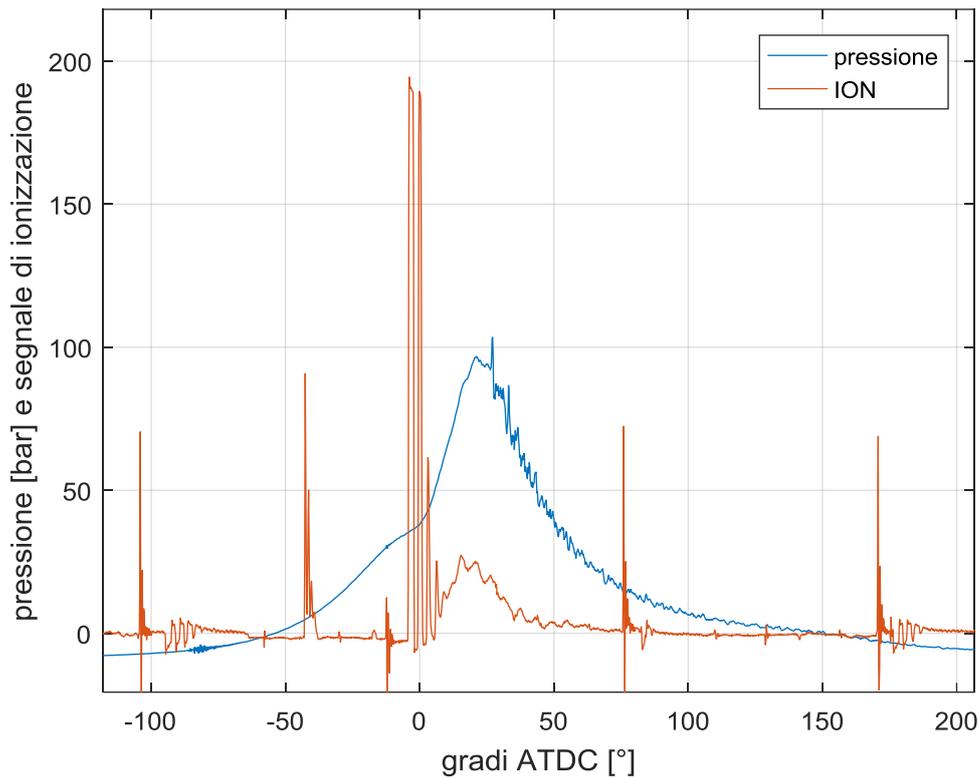


Figura 5.3

È possibile definire anche un indice “integrale” IKI (*Ion Knock Integral*).

5.2 Utilizzo del segnale ION

Come abbiamo detto il segnale ION è disponibile anche in vettura.

Ad oggi, il suo utilizzo è limitato a strategie di protezione dalla detonazione, usando indici simili a quelli che ho definito poco sopra. Viene implementato cioè un controllo in closed-loop dell’anticipo, che ha il “permesso” di togliere gradi di anticipo se viene rilevato un livello di detonazione oltre la soglia di accettabilità.

In fase di calibrazione si ricava il MAPO sulla base della traccia di pressione, e si definisce un indice basato sul segnale ION in modo che sia il più possibile correlato con l’indice MAPO. La correlazione deve evidentemente valere per vari livelli di detonazione.

In questo senso l’utilizzo del segnale ION trova applicazione soprattutto in motori molto frazionati, in cui l’utilizzo di accelerometri risulta difficoltoso.

In vettura il segnale ION viene elaborato da una centralina dedicata, che comunica con la ECU.

Ma le possibilità offerte dall'analisi del segnale ION sono numerose, e la ricerca si sta molto concentrando su questo argomento.

Ad esempio, il segnale ION può essere usato per implementare una strategia di controllo dell'anticipo cilindrico con l'obiettivo di massimizzare le prestazioni (per esempio inseguendo un target di CA50MFB), e quindi non necessariamente solo con una strategia protettiva.

Ancora, si potrebbe implementare un controllo del titolo cilindro per cilindro, o un algoritmo di diagnosi del misfire.

Conclusioni e sviluppi futuri

Il mio lavoro di tesi ha portato al completamento dell'algoritmo ideato per l'analisi dei dati a disposizione e alla realizzazione dei codici necessari per il suo funzionamento. Lo sviluppo dell'interfaccia grafica ha soddisfatto i requisiti fondamentali di essere uno strumento *efficiente e user-friendly*. Ho testato il funzionamento della stessa su più set di dati, ottenendo buoni risultati.

Anche il codice di sincronizzazione svolge "bene" il suo compito, ma può presentare ancora qualche criticità di funzionamento nella prima fase di calcolo, cioè quando vengono sincronizzati i CycleData e i CustomData.

Rimane poi il problema della forte distorsione del segnale INCA, che purtroppo non sono riuscito a compensare del tutto. Allo stato attuale si realizza un errore massimo dell'ordine di 5-6 cicli di sfasamento, mentre idealmente si vorrebbe che i segnali fossero sincronizzati perfettamente, con la precisione di un ciclo. Un'attività futura di miglioramento del codice potrebbe quindi focalizzarsi su questo aspetto.

In generale la metodologia proposta è matura, affidabile ed efficiente, e costituirà dunque un valido supporto alle future attività di analisi dati previste dal progetto di ricerca.

L'esperienza di questi mesi ha permesso al sottoscritto di approfondire la conoscenza del software Matlab, con particolare riferimento all'ambiente di sviluppo per le interfacce grafiche.

Ho inoltre acquisito maggiore familiarità con l'ambiente sala prove motore e con le problematiche relative all'acquisizione e al trattamento di ingenti quantità di dati sperimentali.

Bibliografia

- **Heywood, J.B.** "Internal combustion engines fundamentals".
- **Cavina, N.** Appunti del corso integrato "Propulsori termici e ibridi per autoveicoli M".
- **Minelli, G.** "Motori endotermici alternativi".
- **Bianchi, G. M.** Appunti del corso "Fluidodinamica dei motori a combustione interna M".
- **Klein, M., Eriksson, L.** "A Specific Heat Ratio Model for Single-Zone Heat Release Models".
- **Rojo, N.** Tesi di laurea magistrale
- **Draper, C.S.** "The physical effects of detonation in a closed cylindrical Chamber".