

Studio sull'applicazione della tecnica Sporadic MPC e modelli di simulazione connessi

a cura di Alberto Leva
DEIB, Politecnico di Milano

Sommario

Questo report riguarda una tecnica di controllo di recente introduzione, detta Sporadic Model-based Predictive Control (S-MPC in breve), che mira a conseguire i vantaggi del controllo predittivo model-based (Model Predictive Control, MPC) senza dover risolvere il problema di ottimizzazione in esso coinvolto ad ogni passo di controllo, donde l'aggettivo “sporadico”. In particolare, in questo lavoro la tecnica S-MPC viene accoppiata a una libreria di simulazione Modelica, ottenendo una sua implementazione funzionale che può essere utilizzata per provarla su modelli di complessità e dimensioni ingegneristicamente realistiche. Come prodotto aggiuntivo rilevante del lavoro, sono stati creati e testati modelli Modelica che traggono vantaggio dalla presenza di loop di controllo "periferici" installati in buona parte dei sistemi energetici e ancor più specificamente - dato il contesto del progetto - negli edifici, nell'ottica da un lato di facilitarne la parametrizzazione e dall'altro di ottenere un'alta efficienza computazionale. Queste due proprietà sono infatti particolarmente interessanti in un contesto in cui l'ottimizzazione basata sulla simulazione da un lato svolge (e ci si augura sempre più svolga) un ruolo rilevante e però dall'altro lato deve essere utilizzabile da parte di persone che sono più esperte del campo applicativo che delle tecniche da impiegarsi.

1. Introduzione e contesto

Questo report tratta dell'applicazione della tecnica Sporadic MPC (nel seguito abbreviata in S-MPC). Il componente principale di un sistema di controllo basato su S-MPC è un ottimizzatore e; per quanto ciò non sia tecnicamente necessario, l'interesse di S-MPC proviene largamente dal fatto che in molti casi la funzione di costo non è disponibile in forma chiusa ma dev'essere ottenuta per simulazione. Dato che questo è spesso computazionalmente oneroso, si vorrebbe che il modello utilizzato a questo scopo - chiamato nel seguito "modello di controllo" - da un lato fosse il più semplice possibile ma dall'altro potesse assomigliare il più possibile a quelli utilizzati in ingegneria, per evidenti ragioni di coerenza nel set di dati relativo a un progetto.

Nel lavoro presentato in questo report si studia l'applicazione di S-MPC e si focalizza lo studio in particolare su questo aspetto specifico del problema, dato che il peso computazionale gioca un ruolo dominante sia nel funzionamento di S-MPC sia per l'accettabilità della tecnica. Viene qui proposta una soluzione che si basa sulla presenza di sistemi di controllo periferici, impiegando le loro specifiche di tuning - e le conseguenti parametrizzazioni del loro controllori - per ottenere parte del modello di controllo in forme più semplici di quelle che originano per esempio da un approccio di primo principio. Il miglioramento ottenuto nell'efficienza della simulazione viene quantificato - ancora in via preliminare ma su modelli di complessità non banale - e testimonia l'utilità dell'idea. In secondo luogo, proseguendo nella stessa linea di lavoro, viene qui proposto un ambiente di simulazione per testare soluzioni di controllo S-MPC basate su modelli "realistici" nel senso appena tratteggiato.

2. Analisi della letteratura

In questa sezione viene presentata una breve rassegna della letteratura. Non si pretende ovviamente alcuna esaustività, si intende solo fornire un quadro storico di come è stato raggiunto lo stato dell'arte attuale, e quindi giustificare le proposte formulate in questo lavoro.

2.1 Modellazione e simulazione per il controllo

Tra le tecniche di controllo atte a ottimizzare il rendimento energetico degli edifici, quelle di tipo predittivo basate su modello (Model Predictive Control, MPC) sono considerate tra le migliori, grazie per esempio alla loro capacità di utilizzare il programma di occupazione e le previsioni meteorologiche per un controllo "ottimo". In particolare, in letteratura sono state introdotte funzioni di costo specifiche per minimizzare il consumo di energia mantenendo il comfort termico nell'edificio [22].

L'utilizzo di MPC nei sistemi HVAC (Heating, Ventilation and Air Conditioning) è stato trattato in diversi lavori. In [1] viene studiata l'implementazione di MPC per un impianto ad

acqua refrigerata. Una strategia di controllo predittivo che utilizza un modello di previsione della temperatura dell'aria esterna è stata studiata in [13] per i sistemi di riscaldamento a pavimento radiante con funzionamento intermittente. I risultati sperimentali riportati in quel lavoro mostrano che la strategia di controllo predittivo può risparmiare tra il 10% e il 12% di energia, durante i mesi invernali, rispetto alle strategie di controllo convenzionali. In [38] gli autori descrivono il test di MPC per un sistema di riscaldamento, mentre un caso su edificio reale (a Praga) è descritto in [32]. Tra le conclusioni di questi e molti altri lavori, vale la pena notare che le strategie di controllo predittivo forniscono potenziali benefici per la climatizzazione efficiente degli edifici dal punto di vista energetico, anche perché sono naturalmente inclini a problemi multi-variabili e all'uso coordinato di diversi attuatori, tra cui persiane automatizzate, elementi vetrati regolabili, illuminazione e così via. Inoltre, gli approcci predittivi si prestano naturalmente all'integrazione di previsioni meteorologiche nonché di disponibilità e costi per le varie fonti energetiche, con evidenti vantaggi.

Restando sul fronte HVAC, quando si tratta di valutare le prestazioni termiche in modo abbastanza accurato, un ruolo sempre più rilevante è rivestito dai modelli che rilasciano l'ipotesi che lo stato termodinamico e di moto dell'aria sia spazialmente uniforme. In particolare, sono molto importanti i modelli di flusso d'aria che tengono conto dello stato termodinamico dell'aria in funzione di alcune coordinate spaziali. Questi modelli possono aiutare ad esempio a mantenere un ambiente termicamente confortevole, ottenendo una temperatura e/o una distribuzione dell'umidità più vicine a quelle desiderate e/o aiutando nella scelta del posizionamento migliore per i sensori.

In [3] è descritto un tentativo di affrontare il problema del movimento dell'aria e la previsione delle prestazioni di ventilazione in modo semplice e computazionalmente poco oneroso, introducendo modelli "sub-zonali". L'idea è quella di dividere il dominio spaziale in una griglia "grossolana" di volumi e quindi applicare relazioni semplificate per il calcolo del campo del flusso d'aria attraverso alcuni strumenti come CMOIS [35]. Nel 2003 il lavoro [28] ha proposto il miglioramento del comportamento dei modelli in alcuni casi come la convezione forzata, dove le carenze dell'approccio discusso nel lavoro precedente erano considerevoli. Tuttavia, gli autori non hanno potuto evitare una potenziale perdita di generalità e applicabilità, poiché il suddetto miglioramento si basava su presupposti sperimentali.

Un approccio alternativo che ha ottenuto buoni risultati - e quindi ampiamente adottato negli anni successivi - è discusso in [11]. Gli autori hanno rappresentato un modello di turbolenza particolarmente adatto per la simulazione energetica degli edifici. In tale modello l'utilizzo di mesh abbastanza grossolane, naturalmente connesso a una semplificazione che si accompagna a un aumento della potenza di calcolo, hanno amplificato la penetrazione dei codici di fluidodinamica computazionale (Computational Fluid Dynamics, CFD) nella simulazione edilizia [10]. Questo tipo di approccio è ancora utilizzato dal ben noto software Energy Simulation (ES) che utilizza la co-simulazione per accoppiare tutti i modelli di edificio insieme a un codice CFD che simula l'aria contenuta. Negli ultimi anni, grazie

all'aumento delle prestazioni dei computer nonché all'introduzione di nuove architetture, sono stati studiati nuovi approcci. Un lavoro rilevante in questo contesto è la Fast Fluid Dynamics (FFD) proposta in [44]. FFD utilizza un approccio "semi-lagrangiano", cioè parallelizzabile in modo nativo, quindi può essere notevolmente accelerato se implementato su architetture parallele come quelle GPU (Graphic Processing Unit). Gli stessi autori hanno anche presentato una versione migliorata dei loro modelli in [45], in cui la diffusione numerica indesiderata è stata ridotta mediante tecniche di discretizzazione di ordine superiore [7].

2.2 Controllo predittivo

Recenti studi sull'applicazione di MPC agli edifici hanno condotto ad affermare che il potenziale di risparmio energetico di questa tecnica di controllo è di circa il 40% rispetto alle strategie di controllo convenzionali. Si deve tuttavia menzionare che i risultati della ricerca dipendono da studi di simulazione e che questi sono soggetti alle numerose assunzioni che si fanno [14].

A un livello molto generale, l'utilizzo più tipico di MPC nei controlli degli edifici è riassunto nella Figura 2.1.

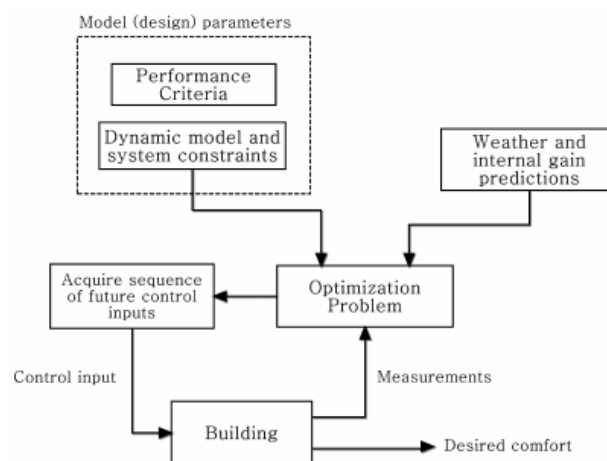


Figura 2.1 : Uso tipico di MPC nei controlli degli edifici.

MPC è un mezzo ampiamente utilizzato nell'industria per affrontare grandi problemi di controllo vincolato a più variabili. L'obiettivo principale è ridurre al minimo su un orizzonte futuro un criterio di prestazione (funzione di costo, cifra di merito, Key Performance Indicator o KPI) proveniente da un sistema controllato che potrebbe essere soggetto a vincoli sugli input e sugli output, dove il comportamento futuro è calcolato secondo un modello del sistema in oggetto. Un controllore MPC utilizza cioè i modelli e le misurazioni sul campo per calcolare i movimenti futuri nelle variabili indipendenti in funzione delle azioni applicate (anch'esse nel futuro e quindi incognite del problema), in modo da minimizzare la funzione di

costo nel rispetto di tutti i vincoli sulle variabili indipendenti e dipendenti. Un controllore MPC invia quindi questa serie di variazioni di variabili indipendenti ai controllori periferici interessati, tipicamente come set point, in modo che essi li applichino (in questo aspetto S-MPC costituisce un'eccezione, come si vedrà in dettaglio nel seguito). MPC, nato all'inizio degli anni ottanta in raffinerie di petrolio, impianti chimici e altre industrie di processo [36], è oggi in realtà divenuto una vasta classe di metodi, accomunati dal calcolare il segnale di controllo riducendo al minimo la funzione obiettivo soggetta a vincoli [26]. La minimizzazione viene eseguita in maniera iterativa su un orizzonte di ottimizzazione finito per acquisire N predizioni in avanti del segnale di controllo che portano al criterio minimo soggetto a tutti i vincoli. Tuttavia, se questi controlli futuri venissero applicati brutalmente come calcolati, ci sarebbero problemi di robustezza significativi, poiché in realtà non ci sarebbe alcun feedback; anche le garanzie di stabilità sarebbero problematiche. Questo è il motivo per cui MPC tradizionalmente - e quasi ovunque - si basa sul cosiddetto approccio "receding horizon", che può rimuovere molti di questi svantaggi. L'idea di tale approccio è che solo il primo campione di controllo proveniente dalla strategia di ottimizzazione viene applicato al sistema controllato e quindi l'intera procedura viene ripetuta nel passo successivo, con l'orizzonte di previsione spostato in avanti appunto di un passo; l'iterazione di tutto questo conduce facilmente a uno sforzo computazionale potenzialmente rilevante [33].

In alcuni riferimenti come [34-43] è stata discussa la stabilità del controllo receding horizon vincolato, mentre in [8] è stato introdotto un altro approccio utilizzando tecniche di controllo robusto. Al fine di impiegare il concetto di controllo predittivo nei sistemi HVAC sono stati fatti diversi tentativi, si vedano per esempio [19, 29, 12, 10, 25]. Il progetto OptiControl, sempre a titolo di esempio, offre una visione articolata in un'ottica di controllo ottimo dell'edificio. OptiControl fornisce anche un'ampia gamma di riferimenti, oltre ai risultati che mostra. Vale infine la pena ricordare che esiste anche un altro progetto che prevede il controllo predittivo dell'accumulo di energia termica nel campus della UC-Berkeley che viene chiamato "controllo predittivo degli edifici in rete".

2.3 Retrospettiva e motivazione

Senza dubbio MPC è una strategia di controllo estremamente promettente per la gestione energetica degli edifici, tuttavia il suo uso richiede spesso di risolvere un problema di ottimizzazione molto complesso, in quanto per ottenere il meglio dall'approccio i modelli semplici - per lo più LTI - che possono fornire la legge di controllo in forma chiusa, semplicemente non sono applicabili. Sono necessarie soluzioni per limitare il carico computazionale e S-MPC è una possibilità praticabile. È quindi opportuno mettere in pratica l'approccio appena citato, in modo da fornire qualche proof of concept per il suo funzionamento sulla base di modelli e problemi di dimensioni ragionevoli.

3. L'approccio seguito

In questa sezione viene innanzitutto affermato il problema affrontato, in modo tale da fornire allo stesso tempo un breve riepilogo delle basi dell'S-MPC. Quindi si discute la scelta degli strumenti utilizzati per comporre la soluzione proposta, evidenziando di conseguenza le esigenze in merito ai modelli di simulazione e all'architettura complessiva di implementazione del controllo.

3.1 Problema e background su S-MPC

L'approccio MPC classico, come anticipato, consiste nell'utilizzare un modello del sistema controllato per prevedere l'evoluzione futura del sistema. Ad ogni passo di controllo, un problema di controllo ottimo ad anello aperto viene risolto su un orizzonte finito. Solo il primo campione della sequenza del segnale di controllo ottimo viene applicato al sistema controllato. Nel passo successivo, un nuovo problema di controllo ottimo, basato su nuove misurazioni, viene risolto su un orizzonte spostato.

Considerando un sistema in condizioni nominali e senza disturbi imprevedibili, si potrebbe però pensare di applicare il vettore dei segnali di comando ottimali non solo per il passo temporale successivo, ma per l'intero orizzonte di predizione. In questo modo si può ridurre fortemente il tempo di calcolo richiesto dall'ottimizzazione MPC. Questo porta allo schema di base per il controllo S-MPC, mostrato in Figura 3.1.

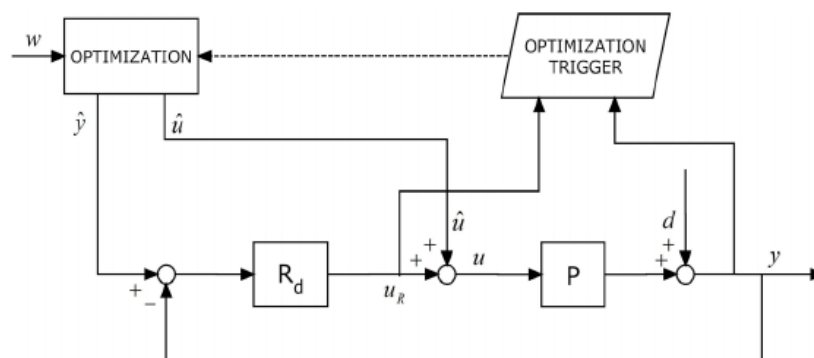


Figura 3.1 : Schema di base per il controllo S-MPC.

In questo schema si usa una tecnica di ottimizzazione per calcolare il vettore u di futuri controlli ottimali, nonché il vettore y contenente la corrispondente traiettoria della variabile controllata, su un orizzonte. Un elemento hold, non mostrato in figura per semplicità, ha il ruolo di acquisire i suddetti controlli futuri e le variabili controllate previste, e applicarli in sequenza, uno per passo di controllo, al loop di feedback nella parte inferiore della figura. Il controllore di questo loop riceve la variabile controllata proveniente dalla predizione (quindi

ottima) come set point e il segnale di controllo (ottimo) come polarizzazione additiva sulla sua uscita, che è il controllo u effettivamente applicato al sistema controllato.

Se il processo sotto controllo è perfettamente catturato dal modello di ottimizzazione e non ci sono disturbi imprevedibili, u fa sì che il sistema controllato emetta esattamente y , e quindi l'uscita di è zero. In caso contrario, tale uscita eserciterà una certa azione, per contrastare sia gli eventuali errori di modellazione sia i disturbi.

3.2 Trigger dell'ottimizzazione

Per forzare S-MPC ad avviare un nuovo processo di ottimizzazione prima dell'applicazione dei tutti campioni di controllo determinati in precedenza all'ultima soluzione dle problema, calcolando così ulteriori controlli futuri e gli output previsti a partire dal passo corrente, abbiamo bisogno di un meccanismo che nella Figura 3.1 è definito "optimization trigger".

Il corretto funzionamento di questo blocco è molto importante, perché più ottimizzazioni vengono risparmiate rispetto al classico approccio receding horizon, maggiore è il risparmio di sforzo computazionale, quindi meglio S-MPC raggiunge il suo obiettivo finale. D'altra parte, tuttavia, se un'ottimizzazione non viene invocata quando ciò sarebbe invece opportuno, la degradazione dei risultati ottenuti rispetto all'MPC puro può essere rilevante, rendendo così S-MPC non efficace o addirittura dannoso.

Per quanto riguarda il blocco trigger di ottimizzazione, ci basiamo sui risultati preliminari di [27] e formuliamo le considerazioni e le proposte riportate nel resto di questa sezione.

Per formalizzare la questione con una notazione il più leggera possibile, si procede in base a una visione intuitiva. Come già detto, a patto che l'esito dell'ultima ottimizzazione corrisponda al comportamento del sistema, il controllore indicato con R_d in figura 3.1 dovrebbe produrre in uscita zero. Un primo possibile criterio è quindi quello di monitorare l'azione di quel controllore - cioè il segnale ed eventualmente l'errore - e intervenire se tale azione oltrepassa una certa soglia in un certo senso, sia da qualificare, che per indicare che “sta facendo troppo rispetto a quello che dovrebbe fare”. Questa idea dà origine a diverse possibilità, come elencato di seguito.

- Un primo modo è utilizzare una delle numerose tecniche proposte in letteratura sui regolatori con autotuning per decidere quando un'operazione di taratura è opportuna, si veda ad esempio [23] - dove vengono monitorati sia la variabile controllata che il segnale di controllo - e i lavori ivi citati. Queste tecniche, oggi ben valutate e che hanno trovato anche alcune applicazioni industriali, hanno però due limiti. Il primo è che sono preferibilmente da applicare quando il set point è costante, il che dati i segnali di riferimento tipici nella maggior parte delle applicazioni legate all'edilizia non dovrebbe essere un problema. Il secondo è che tendono ad essere in qualche modo correlati alla tecnica utilizzata per regolare il controllore, ad esempio perché in qualche modo utilizzano un modello di processo nominale. Nonostante le considerazioni riportate più avanti nel report indichino che questa messa a punto non

ha necessità “strane” e potrebbe essere realizzata con tecniche standard, resta il problema di come sfruttare l'idea appena citata in assenza di informazioni relative alla messa a punto, come sarebbe ad esempio il caso se il metodo proposto doveva essere integrato in uno schema di controllo esistente (tipicamente di tipo decentralizzato).

- Un'altra possibilità è quella di rilevare cambiamenti bruschi e prolungati nel segnale di controllo, il che tuttavia è sensibile a numerose fonti di errore. Questa può essere vista come un'idea di ultima istanza in assenza di altre informazioni, ma non sembra particolarmente promettente.
- In alternativa, si può calcolare l'integrale nel tempo dell'errore e, quando esso supera una soglia, reimpostarne il valore e invocare un'ottimizzazione. Idee simili sono state usate come "rilevamento della variazione di processo" in molti lavori, si vedano di nuovo i riferimenti di [23]. Tuttavia, supponendo che l'obiettivo finale sia ottenere il comfort desiderato risparmiando energia, le deviazioni positive e negative da zero dovrebbero essere ponderate in modo diverso.

Come si può vedere, decidere quale segnale è più rappresentativo della necessità di una nuova ottimizzazione non è un compito banale. Osservato questo, si potrebbe spostare l'attenzione verso l'alto per quanto riguarda il livello di astrazione e monitorare direttamente la funzione di costo da ottimizzare; la *logica* è che detta funzione è la più diretta rappresentante della qualità della soluzione trovata nell'ultimo passaggio di ottimizzazione e, se le previsioni sono corrette, dovrebbe evolversi verso il valore che copre l'intero orizzonte di previsione come derivante dall'ottimizzazione stessa.

Per illustrare un po' più in dettaglio la questione, supponiamo che il problema di ottimizzazione, nelle incognite x di cui si cercano i campioni futuri, sia

$$x^o = \arg \min J(x, \theta) \quad (3.1)$$

soggetto a vincoli non essenziali per questa discussione, dove θ denota qualsiasi altra variabile non soggetta ad ottimizzazione e quindi vista come esogena, e dove J ha una forma del tipo

$$J(x, \theta) = \sum_{\ell=0} N_{opt} J(x(\ell), \theta(\ell)). \quad (3.2)$$

Una volta trovata, si può definire la “funzione di costo nominale corrente” su un orizzonte di lunghezza a partire dal passo corrente come

$$J^o(k) = \sum_{\ell=0} k J(x(\ell), \theta(\ell)). \quad (3.3)$$

Quando il sistema reale funziona, tuttavia, le quantità esogene Θ non hanno il valore nominale ma uno "perturbato", quindi la funzione di costo "corrente *effettiva*" sarà

$$J(k) = \sum_{\ell=0}^{\infty} k J(x(\ell), \theta_{\delta}(\ell)). \quad (3.4)$$

Confrontando $J(k)$ si può quindi vedere se la realtà sta deviando dalle previsioni di ottimizzazione. Più precisamente, come detto in [27] e assumendo che $J > 0$ per costruzione, si può considerare la variazione normalizzata

$$\delta J(k) = \frac{J(k) - J^o(k)}{J^o(k)}. \quad (3.5)$$

Questa è un'idea abbastanza percorribile, in quanto calcolare il valore di una funzione di costo è certamente un compito computazionalmente molto meno oneroso rispetto alla sua ottimizzazione, anche se la suddetta funzione di costo deriva dalla simulazione di un modello dinamico. Tuttavia, anche in questo caso la esiste il problema di come trattare incrementi e decrementi di J rispetto al suo comportamento nominale, poiché i decrementi indicano che le cose stanno andando anche meglio del previsto nell'ultima ottimizzazione, che potrebbe non richiederne affatto un'altra. L'utilizzo delle informazioni raccolte in tali casi, ovvero quando il sistema non si comporta come previsto ma questa discrepanza *migliora* i risultati del controllo, è attualmente una questione aperta.

Riassumendo, il criterio di riattivazione che decide quando eseguire un nuovo problema di ottimizzazione può

- valutare la quantità di energia che il feedback locale fornisce attraverso un indice appropriato, e attivare l'ottimizzazione se l'azione del controllore risulta troppo forte rispetto a una soglia;
- valutare la funzione di costo $J(k)$ nel tempo confrontandola con quella ottimale, e innescare l'ottimizzazione se $J(k)$ sta peggiorando;
- analizzare l'uscita y rispetto al suo riferimento \hat{y} valutando lo scostamento di detto segnale, e anche qui decidere di ottimizzare in base a qualche soglia.

In questo lavoro abbiamo scelto il primo metodo per determinare il criterio di riattivazione e, oltre a questo, per risparmiare il maggior sforzo di controllo possibile.

3.3 Scelta degli strumenti

Per quanto riguarda l'implementazione prevista di S-MPC, ci sono fondamentalmente due *scenari* da considerare. Uno è il setup interamente simulato per testare e valutare una strategia S-MPC, l'altro - non ancora affrontato nel presente documento ma considerato per la scelta degli strumenti - è il vero deployment sul campo di tale strategia.

Per il primo scenario abbiamo bisogno di uno strumento di simulazione che possa trattare facilmente un modello non lineare e multi-fisico, costruito nel modo più simile possibile a quelli per l'ingegneria e che possa essere facilmente strutturato per l'inizializzazione, l'ottimizzazione e così via. Abbiamo anche bisogno di uno strumento di ottimizzazione che possa essere facilmente accoppiato a quello di simulazione. Infine, è necessario un mezzo per riunire i due precedenti in un ciclo operativo funzionale.

Il secondo scenario è analogo, però il modello utilizzato per emulare il sistema - non per l'ottimizzazione - sarà sostituito da quello reale, mentre il modello utilizzato per l'ottimizzazione risiederà nell'architettura di controllo. È quindi necessario che l'ambiente di simulazione possa generare codice eseguibile in modo tale che l'integrazione di tale codice nel tipico sistema di controllo dell'edificio non ponga problemi troppo significativi. Sono state analizzate diverse possibilità, studiando a fondo la documentazione degli strumenti ed eseguendo anche varie prove; i risultati sono riassunti (in estrema sintesi) nel seguito.

3.3.1 GenOpt

GenOpt è un programma di ottimizzazione sviluppato per le ottimizzazioni di sistema di cui abbiamo bisogno. È progettato per trovare i valori dei parametri di progettazione selezionati dall'utente che riducono al minimo una cosiddetta funzione obiettivo, come il consumo energetico annuale, il picco della domanda elettrica o la percentuale prevista di persone insoddisfatte (valore PPD), portando al miglior funzionamento di un dato sistema. La funzione obiettivo viene calcolata da un programma di simulazione esterno, come SPARK, EnergyPlus, DOE-2, TRNSYS, ecc. GenOpt può anche identificare parametri sconosciuti in un processo di adattamento ai dati.

GenOpt consente di connettersi a qualsiasi programma di simulazione con I/O basato su testo semplicemente modificando un file di configurazione, senza richiedere modifiche al codice. Inoltre, ha un'interfaccia aperta per aggiungere facilmente algoritmi di minimizzazione personalizzati alla sua libreria [30]. Ciò consente di utilizzare GenOpt come ambiente per lo sviluppo di algoritmi di ottimizzazione. Questo è il motivo per cui l'abbiamo scelto come il fulcro della parte di ottimizzazione per la nostra realizzazione S-MPC.

Ciò lascia quindi come principale punto in sospeso la scelta del linguaggio di modellazione e del wrapper software complessivo.

3.3.2 EnergyPlus

Possiamo trovare le radici dell'ambiente di simulazione degli edifici EnergyPlus sia nei programmi BLAST che nei programmi DOE-2. Sia BLAST (Building Loads Analysis and System Thermodynamics) che DOE-2 sono stati sviluppati come strumenti di simulazione ed energia alla fine degli anni '70 e all'inizio degli anni '80. La necessità principale di questo programma è che l'ingegnere progettista o l'architetto adattino le apparecchiature HVAC alle dimensioni appropriate, ottimizzino le prestazioni energetiche, e così via. EnergyPlus è un programma di analisi energetica e simulazione del carico termico. secondo la descrizione di

molti utenti di un edificio dal punto di vista della struttura fisica dell'edificio, dei sistemi meccanici associati, etc., questo programma calcolerà il riscaldamento e il raffreddamento carichi al fine di mantenere i set point di controllo termico, le condizioni in tutto un sistema HVAC secondario e i carichi della bobina e il consumo di energia da parte delle apparecchiature dell'impianto primarie, nonché molti dettagli di simulazione necessari per verificare che la simulazione funzioni come farebbe l'edificio reale [31]. EnergyPlus si integra bene con GenOpt ma non gestisce i modelli a-causali.

3.3.3 Matlab-Simulink

Matlab-Simulink si occupa della progettazione a livello di sistema, simulazione, generazione automatica del codice, test continuo e verifica dei sistemi embedded, tuttavia è uno strumento tradizionale per la modellazione e il suo svantaggio più considerevole è la mancanza di orientamento agli oggetti, quindi il modello deve essere trasportato in forma spazio-stato. Strumenti convenzionali per la simulazione e la modellazione come Matlab-Simulink richiedono modelli causali, quindi in questo caso dobbiamo definire quali variabili sono gli input (cause) e quali sono gli output (conseguenze) dei componenti del modello, quindi questa sarebbe una restrizione spiacevole e significa che piccoli cambiamenti nel modello del sistema o l'adattamento del modello esistente a un sistema simile è un processo che richiede tempo con alto rischio di errori, a meno che il modello originale sia stato progettato tenendo conto di queste particolari estensioni. Un altro svantaggio di Matlab-Simulink è che questo strumento non è in grado di includere la documentazione di un modello in quel modello, quindi l'esecuzione del modello è scollegata dalla sua documentazione, quindi di solito la conseguenza principale sarebbe che la documentazione del modello non viene aggiornata con tutte le ultime modifiche nel modello.

Per ottenere una migliore flessibilità nella simulazione e anche nella fase di modellazione, oltre a ottenere un notevole supporto dallo strumento in un ambiente orientato agli oggetti, abbiamo deciso di utilizzare il linguaggio Modelica, accoppiando Genopt e JModelica. L'ultima questione è quindi la scelta del wrapper.

3.3.4 C ++

C ++ è un linguaggio di programmazione generico. È inoltre sviluppato dal linguaggio di programmazione C originale. Il C ++ è un linguaggio di programmazione compilato, in forma libera, multi-paradigma e tipizzato staticamente. Il linguaggio di programmazione C ++ è ampiamente considerato un linguaggio di livello intermedio, in quanto composto da funzionalità del linguaggio sia di alto livello che di basso livello.

3.3.5 Python

Python è un linguaggio di programmazione dinamico, fortemente tipizzato, orientato agli oggetti, multiuso, progettato per essere veloce e per applicare una sintassi pulita e uniforme. Python è un linguaggio di programmazione generico e di alto livello ampiamente utilizzato. La

sua filosofia di progettazione enfatizza la leggibilità del codice e la sua sintassi consente ai programmatori di esprimere concetti in meno righe di codice rispetto a quanto sarebbe possibile in linguaggi come C ++ o Java. due volte più veloce dello sviluppo con C ++ o Java Python supporta più paradigmi di programmazione, tra cui programmazione orientata agli oggetti, imperativa e funzionale o stili procedurali. È dotato di un sistema di tipo dinamico e di gestione automatica della memoria e dispone di una libreria standard ampia e completa.

Come si può vedere, da molti punti di vista Python è molto diverso da C ++. Python è considerato più pulito e diretto, con enfasi sulla leggibilità del codice. Un altro vantaggio di Python è che il suo codice è piuttosto più breve della maggior parte degli altri linguaggi di programmazione. Ciò consente ai programmatori di esprimere concetti con un numero inferiore di righe di codice rispetto a C o C ++. Il linguaggio di Python fornisce costrutti. Questi costrutti hanno lo scopo di consentire programmi chiari su piccola e grande scala.

In conclusione, la scelta che si consiglia per questo lavoro è JModelica più Genopt, con un wrapper Python per gestire l'intero sistema. Le sezioni seguenti trattano i modelli di simulazione Modelica e l'uso di detti strumenti per assemblare la soluzione complessiva.

4. Modelli di simulazione

In questa sezione ci occuperemo dei modelli di simulazione. Ci concentriamo qui sulle questioni specifiche sollevate dalla connessione dei modelli realizzati ad un ottimizzatore in un vista su S-MPC. Per inciso, l'argomento così trattato fornisce un'ulteriore motivazione per la scelta di un paradigma orientato agli oggetti, ma viste le peculiarità di questa motivazione, preferiamo che essa emerga da questo capitolo in modo "spontaneo".

4.1 La libreria Modelica EEB

La modellazione e la simulazione sono sempre state una questione importante nell'ottimizzazione dell'efficienza energetica degli edifici [21-37]. La sfida fondamentale è affrontare efficacemente la natura intrinsecamente multifisica del problema [42]. Il funzionamento dei sistemi di controllo, le azioni degli abitanti ed infine i fenomeni eterogenei (tipicamente termici, idraulici, elettrici ecc.) Impattano sulla prestazione energetica dell'edificio; in altre parole, la prestazione energetica è definita dall'interazione di tutti i fenomeni menzionati [41], il che rende ulteriormente complicato lo scenario . Praticamente la totalità degli strumenti ingegneristici e delle pratiche progettuali si distinguono

- l'edificio *stricto sensu* , ovvero muri, finestre e porte;
- le portate d'aria contenute che si dividono in zone;
- i sistemi di automazione e controllo;

- il sistema HVAC;
- le fonti e i pozzi energetici per l'utilizzo dell'edificio.

Per un tale scenario le problematiche molto diverse poste dai vari sottosistemi sono la principale difficoltà. Ad esempio, passando direttamente alla questione centrale, i blocchi orientati sono ideali per strutturare e rappresentare i modelli del sistema di controllo, ma possono richiedere una rappresentazione a tempo continuo o digitale a seconda dello scopo della simulazione; al contrario, i modelli HVAC - se realizzati come primo principio come suggerisce la pratica - sono invarianti nel dominio del tempo continuo, ma sono tipicamente zero o al massimo unidimensionali, sebbene modelli di fenomeni come un muro o un volume d'aria (ciò che accade nei *continui* di solito non può evitare le rappresentazioni 3D, si veda [6] come esempio).

In conclusione di quanto abbiamo appena discusso brevemente qui e ricordando anche [6], è difficile ideare modelli che affrontino tutti i fenomeni necessari, e possano essere organizzati in modo modulare a vantaggio della loro costruzione, parametrizzazione e manutenzione. Per dimostrarlo, è sufficiente gettare uno sguardo alla letteratura relativa all'edilizia, e in particolare all'argomento e al flusso di ricerca chiamato "simulazione dell'intero edificio": questioni metodologiche e ingegneristiche sono discusse, ad esempio in [15, 24] mentre le difficoltà affrontate sono ben evidenziate da [18-40].

Negli ultimi anni al Politecnico di Milano è stata creata la libreria EEB (Energy-Efficient Buildings), che viene attivamente sviluppata e mantenuta. Per ragioni evidenti, l'idea è di avere il materiale S-MPC compatibile e integrato con quella libreria. Evitando problematiche non centrali in questo report, la preoccupazione principale per raggiungere questo obiettivo è la rappresentazione dei sistemi di controllo, in quanto anche quelli a tempo continuo possono a volte causare rallentamenti della simulazione sebbene riferiti a loop periferici, che sono di poca - se del caso - rilevanza per lo strato MPC sulla cui realizzazione è centrata questo lavoro.

L'idea sfruttata nel seguito, quindi, è se - e in caso affermativo, come - si possa trarre profitto a livello di ottimizzazione, dal fatto che alcuni degli oggetti modellati hanno i propri controlli locali installati.

4.2 Modelli per elementi controllati

L'idea di base per questo particolare tipo di modelli è che quando qualche elemento ha un controllo locale, la variabile controllata può essere calcolata filtrando il set point ed i disturbi considerati - che in simulazione sono segnali disponibili - attraverso funzioni di trasferimento dipendenti, per l'approssimazione a livello di sistema richiesta, solo dalle specifiche di controllo. Una volta calcolata la variabile controllata, è spesso possibile ricostruire - parzialmente o totalmente, direttamente o tramite osservatori - le variabili di stato del modello dell'elemento, e quindi altre grandezze relative ad esempio al consumo energetico. Ciò potrebbe consentire di non risolvere modelli complessi, inclusi ad esempio i calcoli delle

proprietà delle sostanze, e limitare al minimo (o addirittura eliminare totalmente) le discontinuità e gli eventi come quelli causati dalla gestione della saturazione.

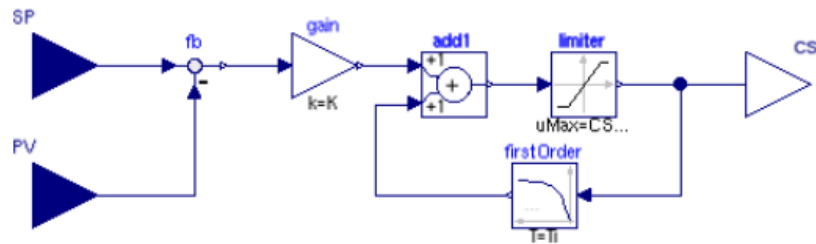


Figura 4.1 : Antiwindup PI realizzato in Modelica.

In modo abbastanza intuitivo, una prima idea è creare blocchi controllore, nel dominio del tempo continuo, privi di elementi che generano eventi come le clausole `if` o `max/min`. Considerando ad esempio il PI antiwindup rappresentato come diagramma Modelica in figura 4.1, si può semplicemente sostituire il blocco "limiter" con uno che fornisce un'uscita continua. Un modo per farlo è definire una funzione a gradini continua, ad esempio utilizzando la tangente iperbolica; un esempio è la funzione `fc01` nel listato seguente.

```

1 function fc01
2   input Real u;
3   input Real p=1000;
4   output Real y;
5   algorithm
6     y := 0.5 + 0.5 * tanh(p * u);
7 end fc01;

```

Sulla base di ciò, è abbastanza semplice ottenere una funzione di saturazione continua; un esempio è dato nel listato seguente.

```

1 function fcsat
2   input Real u;
3   input Real umin=0;
4   input Real umax=1;
5   input Real p=1000;
6   output Real y;
7   protected
8     Real cmin, cmax;
9   algorithm
10    cmin := fc01(u - umin, p);
11    cmax := fc01(u - umax, p);
12    y := cmax*umax+(1-cmax)*(cmin*u+(1-cmin)*umin);
13 end fcsat;

```

Usando la funzione fcsat nel listato, sostituire il blocco di saturazione in figura ? evita gli eventi di stato generati dal PI, ma richiede comunque di risolvere il modello dell'intero loop - possibilmente includendo un sistema controllato complesso - per calcolare variabili il cui comportamento dovrebbe essere dominato da un controllore progettato in modo appropriato. Per gli studi qui affrontati, che possono richiedere un numero enorme di simulazioni, rimane la domanda (più ambiziosa) se e come tali calcoli complessi possano essere evitati.

Per illustrare l'idea che proponiamo come soluzione pratica, si consideri il caso più semplice - ma non lontano da alcuni di interesse pratico - in cui un processo di primo ordine, privo di ritardi è controllato da un PI sintonizzato in modo da prescrivere un certo circuito chiuso tempo di assestamento. Scrivendo la funzione di trasferimento del processo come

$$P(s) = \frac{\mu_P}{1 + sT_P} \quad (4.1)$$

e indicando il detto tempo di assestamento richiesto con , il PI viene prontamente sintonizzato (mediante cancellazione) per essere

$$R(s) = \frac{5}{sT_a} \frac{1}{P(s)} = \frac{5}{\mu_P T_a} \frac{1 + sT_P}{s}, \quad (4.2)$$

dove la frequenza critica è stata calcolata, secondo la pratica comune, come . Questo è scritto nel modulo ISA come

$$R(s) = K \left(1 + \frac{1}{sT_i} \right) \quad (4.3)$$

con

$$K = \frac{5T_P}{\mu_P T_a}, \quad T_i = T_P. \quad (4.4)$$

In condizioni nominali, alle quali faremo riferimento per stimare le variabili richieste senza simulare il loop, la funzione di trasferimento dal set point $w(t)$ alla variabile controllata $y(t)$ è

$$T(s) = \frac{R(s)P(s)}{1 + R(s)P(s)} = \frac{1}{1 + sT_a/5}, \quad (4.5)$$

mentre quello da un disturbo del carico $d(t)$ entrante in modo additivo all'ingresso del processo e $y(t)$ è

$$F(s) = \frac{P(s)}{1 + R(s)P(s)} = \frac{s\mu_P T_a/5}{1 + s(T_P + \frac{T_a}{5}) + s^2 \frac{T_P T_a}{5}}. \quad (4.6)$$

Supponiamo ora di conoscere sia la parametrizzazione del controllore che le specifiche da cui proviene. In questo caso

$$\mu_P = \frac{5T_i}{KT_a}, \quad T_P = T_i, \quad (4.7)$$

donde

$$F(s) = \frac{sT_i/K}{1 + s(T_i + \frac{T_a}{5}) + s^2 \frac{T_i T_a}{5}}. \quad (4.8)$$

Per mezzo di (4.5) e (4.8) possiamo quindi stimare $y(t)$ a partire da $w(t)$ e $d(t)$, senza calcolare i modelli del loop, a condizione che sia i requisiti di controllo e di i conseguenti parametri del regolatore presente in quel loop sono noti - un aspetto che sottolineiamo, per quanto possa sembrare ovvio, dal momento che in alcune situazioni quest'informazione non risulta invece facilmente reperibile, evidenziando l'importanza di una procedura di sintesi di controllo formalizzata e soprattutto ben documentata.

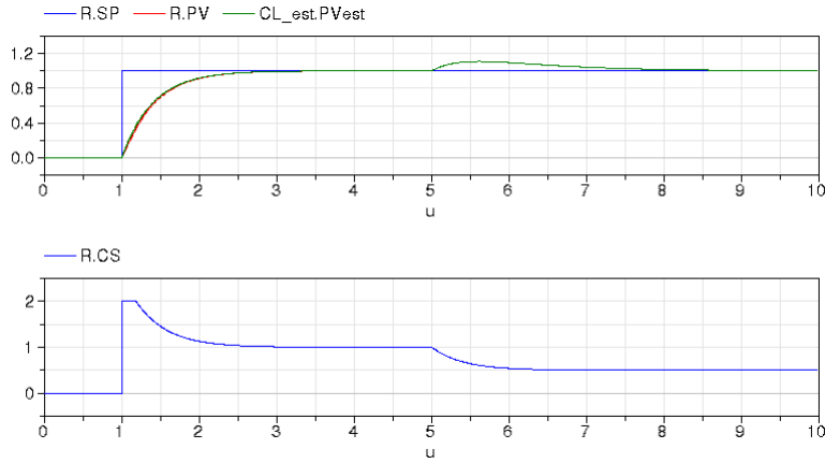


Figura 4.2 : Stima di variabili controllate - esempio 1.

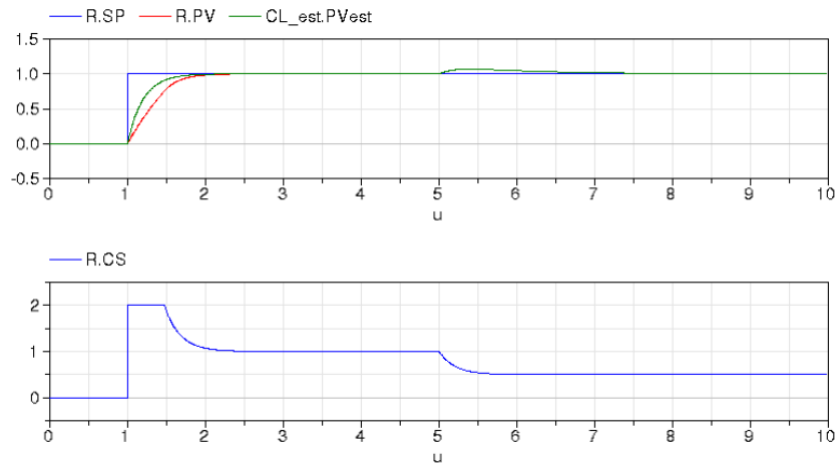


Figura 4.3 : Stima di variabili controllate - esempio 2.

Per indagare l'accuratezza della stima appena abbozzata, mostriamo un primo paio di esempi, in cui ci concentriamo sulla stima della variabile controllata. Il processo è il medesimo (i dettagli sono inessenziali per questi esempi) e per esso vengono sintetizzati due controllori, con un tempo di assestamento T_a richiesto rispettivamente di 2 e 1. Uno scalino unitario di set point viene applicato al tempo $t = 1$ e un disturbo di carico, anch'esso a scalino unitario, si verifica a $t = 5$.

La figura 4.2 riporta i risultati per $T_a=2$, mentre la figura 4.3 quelli per $T_a=1$. In entrambi i casi, il grafico in alto mostra il set point, la variabile controllata effettiva e quella stimata come illustrato sopra. Come si può vedere, c'è qualche errore di stima, causato - come ci si può aspettare - dalle saturazioni di controllo, che intrinsecamente non sono rappresentate dalla funzione di trasferimento usata per calcolare la stima di $y(t)$. Tuttavia, fintanto che le variazioni di set point non provocano saturazioni eccessive, e fintanto che i disturbi di carico

non sono così bruschi da indurre lo stesso fenomeno, le stime ottenute con il semplicissimo approccio proposto sono abbastanza soddisfacenti, e consentono di ipotizzare con sicurezza che per qualsiasi scopo per i quali risultano adeguati, come gli studi di sistema affrontati in questo lavoro, si può evitare la simulazione del modello "completo" o meglio "fisico" del loop, contenente una descrizione completa - e a volte complessa - del sistema controllato.

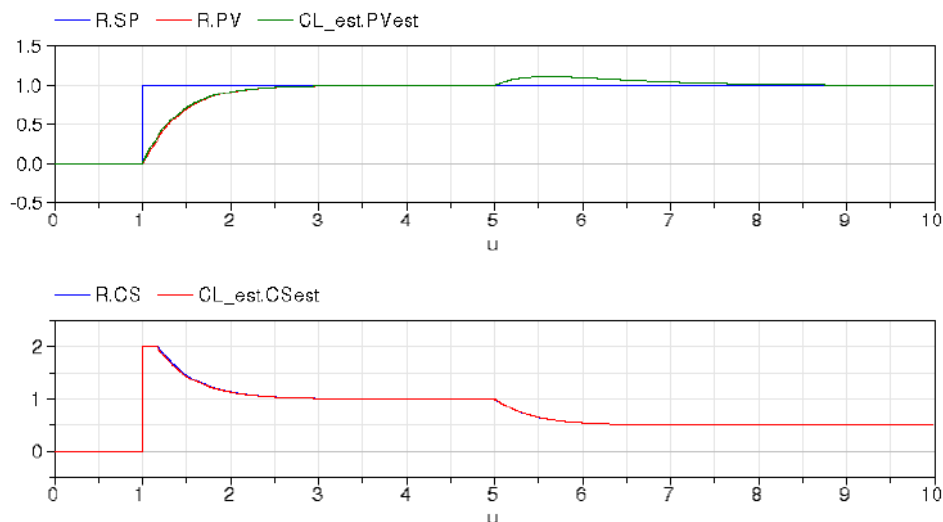


Figura 4.4 : Variabile controllata e stima del segnale di controllo - esempio 1.

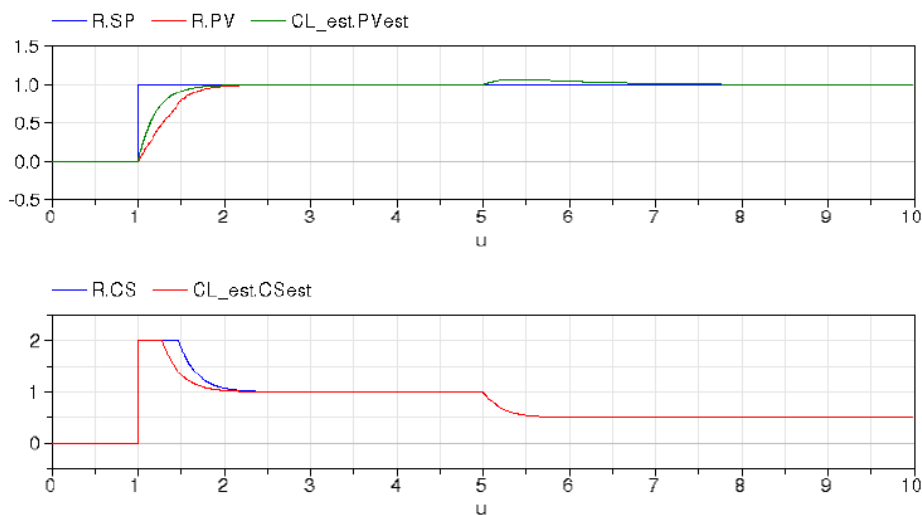


Figura 4.5 : Variabile controllata e stima del segnale di controllo - esempio 2.

Per quanto riguarda la stima del segnale di controllo, le cose sono un po' più complicate. L'idea più semplice è osservare che, denotando il suddetto segnale con $u(t)$,

$$Y(s) = P(s)(U(s) + D(s)), \quad (4.9)$$

cosicché

$$U(s) = \frac{1}{P(s)}Y(s) - D(s). \quad (4.10)$$

Si potrebbe quindi stimare $u(t)$ con un'inversa approssimata del processo, per esempio

$$\Pi(s) = \frac{1}{\mu_P} \frac{1 + sT_P}{(1 + sT_P/N)^2}, \quad (4.11)$$

dove $N > 1$ è un parametro che bilancia l'accuratezza della soluzione rispetto alla stiffness del sistema approssimativo ottenuto e può essere reso dipendente dai requisiti di controllo e dai parametri del controllore in base a (4.7). La scelta di un modello strettamente proprio, con il grado relativo minimo di uno, serve ad evitare comportamenti non fisici del segnale di controllo in risposta a scalini (o in generale variazioni brusche) del disturbo di carico.

Tutto sommato, quanto sopra equivale a calcolare la $u(t)$ stimata sostituendo $\Pi(s)$ come dalla (4.11) al posto di $P(s)$ nella (4.10) e sottoponendo l'uscita di quest'ultimo a saturazione. Vale la pena di notare che questa saturazione non fa parte di nessun loop interno a un controllore, come quello che si può vedere in figura 4.1, e quindi ha un impatto molto limitato sulla complessità del modello complessivo.

Le figure 4.4 e 4.5 riportano due esempi di applicazione della tecnica appena proposta, nello stesso caso delle figure 4.2 e 4.3 più sopra. Come si può vedere, le prestazioni ottenute sono abbastanza soddisfacenti, nonostante la semplicità della soluzione.

Un'altra idea per la stima del segnale di controllo è combinarla con quella della variabile controllata, invece di stimare prima quest'ultima e poi ottenere la prima per inversione approssimativa del processo. Ciò richiede di scrivere un modello LTI approssimativo del loop di controllo combinando la descrizione nello spazio di stato del processo

$$\begin{cases} \dot{x}_P(t) &= -\frac{1}{T_P}x_P(t) + \frac{\mu_P}{T_P}u(t) \\ y(t) &= x_P(t) \end{cases} \quad (4.12)$$

e del controllore PI come in figura 4.1, ossia

$$\begin{cases} \dot{x}_R(t) &= -\frac{1}{T_i}x_R(t) + \frac{1}{T_i}u(t) \\ u(t) &= \max\left(u_{min}, \min\left(u_{max}, x_R(t) + K(w(t) - y(t))\right)\right) \end{cases} \quad (4.13)$$

che non elaboriamo fino a una forma di stato completamente ben posta poiché la descrizione sopra può essere espressa direttamente nel linguaggio Modelica nella forma in cui si trova, facendo della questione un puro fatto formale senza conseguenze operative. Per limitare gli

eventi generati, è possibile sostituire ulteriormente gli operatori `max` e `min` con funzioni continue o utilizzare la clausola `noEvent`.

In base all'esperienza acquisita, però, l'idea appena esposta non offre alcun significativo avanzamento rispetto a quella semplice proposta prima, che è quindi quella adottata nei modelli di elementi controllati inclusi nella libreria EEB. Per esemplificare come questi modelli possono essere utilizzati nei nostri studi, presentiamo ora un esempio.

4.3 Modello di un riscaldatore centrale controllato

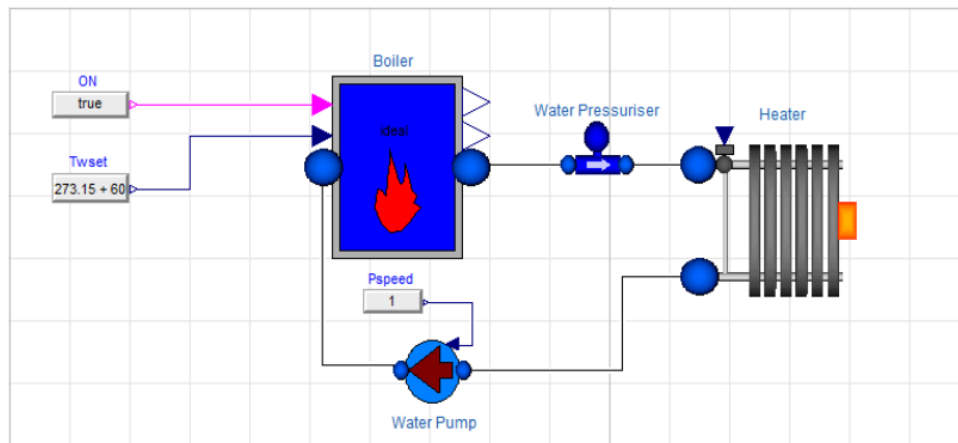


Figura 4.6 : Modello di riscaldatore controllato

Per mostrare un esempio di come viene messa in atto l'idea tratteggiata sopra si consideri lo schema della Figura 4.6, che contiene uno riscaldatore centralizzato per acqua (e un unico utilizzatore per semplicità, ma le cose non cambierebbero in presenza di un carico termico più complesso).

Supponendo che la temperatura di uscita del riscaldatore sia controllata e anche la portata del fluido riscaldante sia regolata, le condizioni di ritorno al riscaldatore sono determinate dal carico termico.

Conoscendo le condizioni di ingresso e uscita del fluido riscaldato, e la portata, è quindi possibile calcolare il calore ceduto al fluido stesso. Di conseguenza, il modello di un riscaldatore di fluido idealmente controllato (omettendo alcune dichiarazioni e definizioni non essenziali qui) è riportato nel listato qui di seguito.

```

1  model IdealControlledFluidHeater
2    ...
3    Media.Substances.SubcooledWater water;
4    parameter SI.Time Tcl = 5
5        "closed-loop T control time constant";
6    parameter SI.Time Thc = 20
7        "intrinsic cooling time constant";
8    parameter SI.Temperature Tstart = 273.15 + 25
9        "initial fluid T";
10   SI.Temperature Tfo(start = Tstart) "outlet fluid T";
11   Modelica.Blocks.Interfaces.BooleanInput ON;
12   Modelica.Blocks.Interfaces.RealInput To;
13   Modelica.Blocks.Interfaces.RealOutput Pc;
14   Modelica.Blocks.Interfaces.RealOutput Ec;
15   equation
16     // no inlet-outlet pressure drop
17     p1      = p2;
18     water.p = p1;
19     // no mass storage
20     w1 + w2 = 0;
21     // output enthalpy
22     hout2 = water.h;
23     // output temperature, controlled
24     water.T = Tfo;
25     if ON then
26       Tfo + Tcl * der(Tfo) = To;
27     else
28       Tfo + Thc * der(Tfo) = water.T;
29     end if;
30     // Power consumption
31     Pc = noEvent(max(w1 * (actualStream(flange2.h) -
32       actualStream(flange1.h)), 0.0));
32     Pc = der(Ec);
33     hout1 = inStream(flange1.h);
34   end IdealControlledFluidHeater;

```

L'applicazione della semplice idea di cui sopra è qui completata con l'introduzione di un comando booleano ON, che quando è falso fa sì che la temperatura di uscita raggiunga quella di ingresso con costante di tempo “intrinseca”. Questa è un'approssimazione piuttosto cruda,

ma influisce su un volume (quello del riscaldatore) piccolo rispetto al volume totale di qualsiasi circuito idraulico che si possa incontrare in casi realistici e non ha un impatto significativo sul consumo di energia (quindi carburante) perché corrisponde intrinsecamente alla situazione con portata zero.

Si possono incontrare, nel contesto modellistico del progetto, molti altri casi come questo, in cui sfruttare la presenza del controllo locale all'interno di qualche elemento può aiutare a risparmiare una notevole quantità di onere computazionale per la simulazione.

5 Controllo

In questa sezione descriviamo la realizzazione di S-MPC come d'interessare per il lavoro presentato. Il diagramma di flusso ad alto livello della procedura è illustrato nel diagramma a blocchi della Figura 5.1 . Possiamo dividere questa procedura in due parti principali:

- il problema dell'ottimizzazione, che utilizza un “modello per l'ottimizzazione” (MO di seguito) per calcolare i controlli ottimi e i corrispondenti uscite ottime predette;
- la parte di controllo, che utilizza un “modello per il controllo” (MC nel seguito) per simulare il sistema controllato e riattivare un'ottimizzazione quando necessario (ovviamente questo non sarebbe presente in un'implementazione reale).

Per utilizzare GenOpt e Modelica, tutte le variabili di ottimizzazione devono essere parametri del modello. Quando viceversa si vuole ottenere, come risultato dell'ottimizzazione, segnali e non parametri, ciò si ottiene utilizzando le "time table" Modelica, decidendo *a priori* la discretizzazione temporale dei segnali - ad esempio, in base al campionamento ritenuto adeguato per un soluzione MPC standard del problema affrontato - e avendo come variabili di ottimizzazione i valori del segnale negli intervalli di tempo così scelti.

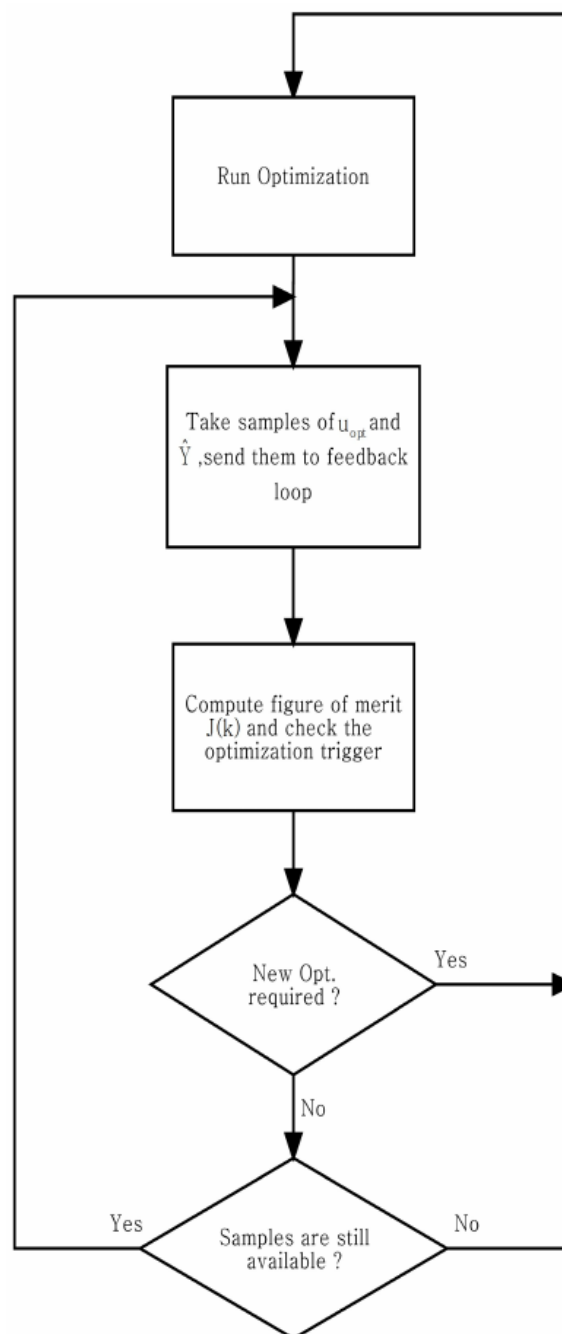


Figura 5.1 : Diagramma di flusso ad alto livello della procedura S-MPC.

Le variabili di ottimizzazione (set point) sono state definite nella parte Modelica dello strumento come “CombiTimeTable”. Il blocco CombiTimeTable utilizza l'interpolazione lineare per fornire $y[:]$ come output del blocco. Sia i valori che i punti temporali sono in una tabella a matrice $[i,j]$. In questa matrice la prima colonna $[:,1]$ include i punti temporali e le altre colonne includono i dati che devono essere interpolati. E' possibile definire quali colonne

vengono interpolate. ad esempio, nel caso di colonne = 2,5 , supponiamo che siano presenti 2 segnali di uscita, la prima uscita sarebbe il risultato dell'interpolazione della colonna 2 e la seconda dell'interpolazione della colonna 5 della tabella. Le proprietà d'interpolazione della tabella possono essere riassunte come segue.

- I punti temporali devono essere rigorosamente crescenti se è richiesta la continuità della derivata prima temporale (ovvero sono esclusi per esempio gli scalini, per cui si mettono in tabella due valori d'uscita diversi allo stesso tempo)
- Le discontinuità sono consentite, fornendo lo stesso punto temporale due volte nella tabella (caso opposto a quello sopra).
- I valori al di fuori dell'intervallo della tabella, vengono calcolati per estrapolazione in base a un apposito parametro (per esempio si mantiene l'ultimo valore).

5.1 Modello per l'ottimizzazione

Anticipando l'esempio della sezione 6, che si riferisce al controllo della temperatura per una stanza in un edificio con sistema HVAC, un esempio di MO (Model for Optimization) è mostrato in Figura 5.2.

Per mettere in pratica la procedura fin qui descritta, dobbiamo eseguire le ottimizzazioni mediante un modello che ci consenta di avere la funzione di costo. Nell'esempio sono presenti due fasce orarie. La prima comprende i valori per modificare lo stato del riscaldatore da acceso a spento e viceversa. La seconda tabella è quella dei set point già scelti come temperatura desiderata in ogni ora per la stanza. Il risultato del Modello è J , ovvero la funzione di costo che deve essere minimizzata per ottenere l'output desiderato. Il passaggio successivo consiste nell'eseguire il modello un'altra volta, ma questa volta invece di valori casuali, con i valori ottimizzati di \hat{u} per ottenere la stima di T_u . Sia \hat{u} e T sono ingressi del modello MC utilizzati per produrre i valori finali per T .

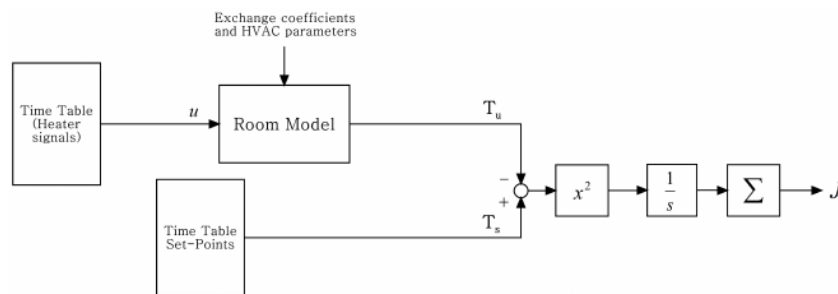


Figura 5.2 : Esempio di modello di ottimizzazione per la procedura S-MPC.

5.2 Modello per il controllo

Anticipando ancora l'esempio della sezione 6, un esempio di MC è mostrato nella Figura 5.3. Aggiungendo il segnale di controllo previsto \hat{u} - che era uno dei risultati del Modello di ottimizzazione già discusso e rappresentato nella Figura 5.2 - al segnale del controllore - che nel nostro caso è stato scelto di tipo PI - e applicando anche il risultato della soluzione del problema di ottimizzazione che fornisce la traiettoria dei segnali ottimizzati T come i set point del regolatore possiamo ottenere l'uscita controllata e ottimizzata T .

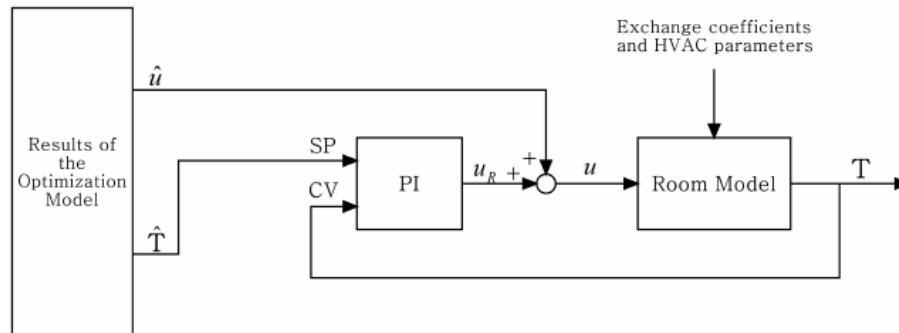


Figura 5.3 : Esempio di modello di controllo (in simulazione) per la procedura S-MPC.

5.3 Scelta del controllore per il loop

La scelta del controllore per il loop ha ovviamente un'influenza sul comportamento dell'S-MPC, quindi spendiamo alcune parole su questo argomento.

Innanzitutto, è necessario scegliere la *struttura* di quel controllore. In linea di principio nessuna indicazione "generale" potrebbe essere fornita se non tenendo conto della fisica del problema di controllo affrontato, ma quando si ha a che fare con controlli HVAC a livello di zona, si può essere ragionevolmente sicuri che le dinamiche incontrate sono prevalentemente ben rappresentabili con un modello dal primo o al più del secondo ordine. La scelta di un controllore PI o PID è quindi adeguata.

Quindi, deve essere considerato il modo in cui è tarato il controllore. Nelle più tipiche applicazioni HVAC, l'obiettivo principale è di tipo "regulatory", ovvero il controllore deve essenzialmente rigettare i disturbi in vista del mantenimento di un set point che per la grande maggioranza del tempo rimane costante. Nel caso di un PI, ciò normalmente equivale a mantenere il tempo integrale il più piccolo possibile compatibilmente con i requisiti in essere sul grado di stabilità - la procedura SIMC [39] è un esempio di taratura che segue questo razionale e che ha incontrato un successo abbastanza rilevante nelle applicazioni.

Nel caso di un set point costante, quindi, l'ottimizzazione segue linee guida abbastanza semplici. Il problema che si discute ora è quello che accade in presenza di un set point fornito

in termini di campioni della variabile controllata ottima come essi sono stati predetti da un ottimizzatore, che è il caso affrontato qui.

Data la complessità del problema è difficile trovare risultati formalizzati, quindi ci affidiamo all'intuizione e puntiamo solo a pratiche sensate. Una prima ovvia osservazione è che il set point per il loop viene sì modificato dall'ottimizzatore, ma ad una cadenza molto più lenta di quella del passo di controllo del loop. Detto set point può infatti essere modificato ad ogni passo, poiché è decisamente improbabile che l'ottimizzazione trovi una soluzione con errore nullo (il che produrrebbe uscita ottima predetta uguale al set point, che questo sì assumiamo quasi sempre costante) a meno che non si utilizzino modelli inaccettabilmente semplici. Tuttavia, questa cadenza di modifica è da supporre in generale lenta.

Con un controllore incentrato sull'inseguimento del set point questa osservazione chiuderebbe la discussione, perché qualsiasi taratura sensata mirerebbe a raggiungere tempestivamente il nuovo valore del set point senza comportamenti indesiderati come sovraelongazioni e/o oscillazioni. Tuttavia ciò non è invece affatto garantito per una taratura centrata invece sulla reiezione dei disturbi, poiché il feedback "trong" necessario si traduce abbastanza facilmente in una reazione troppo nervosa ai cambiamenti del set point. Si potrebbe pensare di sfruttare strutture di controllo a due gradi di libertà, tuttavia una tale funzionalità è tutt'altro che onnipresente nei dispositivi utilizzati per i controlli HVAC periferici.

Fortunatamente, il problema viene automaticamente mitigato dal ruolo che i campioni di controllo futuri ottimali svolgono naturalmente nello schema, cioè quello di fornire una polarizzazione additiva sull'uscita del controllore PI del loop. Questo ruolo dovrebbe corrispondere, almeno in linea di principio, a guidare il loop verso il set point in modo feedforward-centric e quindi non stimolando le dinamiche da set point a output e da set point a controllo, che sono quelle molto probabilmente rese troppo reattive dal ruolo disturbance-centric del loop. Secondo le considerazioni fatte finora, quindi, una buona messa a punto "standard" del controllore di loop secondo le procedure tipicamente usate dovrebbe fornire un controllore intrinsecamente adatto S-MPC, il che è decisamente in fatto favorevole per quanto riguarda l'integrazione di questa tecnica in sistemi che hanno già le loro procedure di sintesi, eventualmente anche automatizzate.

C'è poi un secondo aspetto da tenere in conto. Più aggressivo è il controllore di loop, più è probabile che un meccanismo di attivazione basato su di esso richieda ottimizzazioni, mentre è meno probabile che un meccanismo simile basato sulla funzione del costo "running" faccia lo stesso. Questo perché in tale seconda impostazione qualsiasi non nominalità o disturbo provoca una forte reazione del controllore di loop, che è probabile ripristini rapidamente la funzione di costo. Abbastanza intuitivamente, nel caso di un controllore di loop non aggressivo, ci si aspetta che la situazione si inverta e si abbia prevalenza di trigger basati sulla funzione di costo, anche se non così frequenti come quelli basati sull'energia di controllo nell'altro caso, perché le dinamiche dietro J sono praticamente sempre più lente della scala temporale dell'azione del loop (tanto più se esso è pensato per rigettare i disturbi).

Gli ultimi effetti sono decisamente difficili da descrivere e analizzare in qualsiasi quadro generale ed è ancor più complicato tentare di generalizzare in modo rigoroso le considerazioni del paragrafo precedente. Pertanto, per i test effettuati in questo lavoro, la scelta è sempre stata un PI con taratura "regulatory". Non vi è comunque dubbio che le idee menzionate in questa sezione meritino ulteriori studi.

5.4 Una proposta di applicazione python

La procedura complessiva può essere gestita per semptio da un'applicazione scritta in python, che si occupa di realizzare il diagramma di flusso di Figura 5.1 alternando opportunamente GenOpt (utilizzando MO) e la simulazione del loop S-MPC (utilizzando MC). Un flusso semplificato di esso è mostrato nel listato seguente. Per semplicità mostriamo solo i comandi essenziali e trascuriamo altre parti meno rilevanti.

Per simulare i modelli MO e MC descritti in precedenza, abbiamo utilizzato la piattaforma open source basata su Modelica JModelica. Oltre a questa piattaforma, abbiamo utilizzato il pacchetto PyFMI che ci consente di caricare e interagire con le Functional Mock-Up Unit (FMU).

La struttura del programma è quella di eseguire prima GenOpt utilizzando la capacità di chiamare applicazioni Java in Python in modo che GenOpt produca un file di testo composto da valori ottimizzati.

Il passaggio successivo consiste nel leggere il file di testo citato, arrotondare ogni valore (necessario per GenOpt), trasformare quei valori in stringhe e scriverli in un file di testo di esempio predefinito.

Utilizzando PyFMI si esegui quindi un'altra volta la simulazione del modello MO per ottenere la traiettoria corrispondente delle variabili ottimizzate. PyFMI memorizzerà i risultati in un file di testo in modo che leggendo quel file tramite python possiamo finalmente fornire il set point e la tabella temporale del segnale di controllo del bias aggiuntivo per il modello MC. Fornendo le suddette tabelle abbiamo eseguito la simulazione di MC e verificato il criterio di trigger di ottimizzazione che - come discusso in precedenza - nei test riportati è l'energia di controllo rilasciata dal controllore del loop.

```

1 import subprocess
2 TSH = 0
3 counter = 0
4 x = 1
5 for x in range(True):
6     subprocess.call(["java", "-cp", "C:\genopt\genopt.jar",
7                     "genopt.GenOpt", "D:\JModelica.org-1.17\MC_simple\
8                     MO_simple\optWinXP.ini"])
9     with open("D:\JModelica.org-1.17\MC_simple\MO_simple\
10             OutputListingMain.txt") as f:
11         words = [word for line in f for word in line.split()]
12         print("using f(x) = %s" %(words[297]))
13         a=words[298]
14         b=words[299]
15     ....
16     ....
17     ....
18     from shutil import copyfile
19     copyfile(r'D:/JModelica.org-1.17/MC_simple/MO_simple2/That
20             txt', 'D:/JModelica.org-1.17/MC_simple/That.txt')
21     from pymodelica import compile_fmu
22     from pyfmi import load_fmu
23     import matplotlib.pyplot as plt
24     import numpy as np
25     model_name = 'MC_simple'
26     mo_file = 'MC_simple.mo'
27     VDP_fmu = compile_fmu('MC_simple', compiler_options = {
28         'extra_lib_dirs': 'D:\JModelica.org-1.17\MC_simple'})
29     vdpid = load_fmu(VDP_fmu)
30     res = vdpid.simulate(final_time=86400)
31
32     ... re-triggering criteria generally involve MC only;
33         in the case of interaction with the application,
34         this would be the place to take care of the
35         relative stuff
36
37     x += 1
38     TSH=res["integrator.y"] // re-trigger threshold
39     if TSH>0.2:
40         counter = counter + 1
41         continue
42     else:
43         print(counter)
44         break

```

6 Esempi di simulazione

In questa sezione mostreremo e discuteremo brevemente un esempio di applicazione, in simulazione, per la realizzazione di S-MPC presentata.

L'impianto sotto controllo è costituito da un locale, che scambia calore con l'ambiente esterno ed è sottoposto all'irraggiamento solare. Il locale è dotato di un elemento riscaldante di tipo fan coil, a sua volta collegato a un riscaldatore del fluido di lavoro. Nella stanza è presente un carico termico interno variabile nel tempo per tenere conto di occupanti, elettrodomestici e simili.

L'aria contenuta nell'ambiente è modellata come un volume unico, mentre vengono utilizzate opportune correlazioni di letteratura per descrivere lo scambio convettivo con le pareti sia per la faccia interna che per quella esterna.

Il modello è assemblato con elementi della libreria EEB e adatto all'applicazione python descritta nella sezione 5. Sebbene questo sia ancora un modello abbastanza semplice, il setup così ottenuto permette di testare le applicazioni complete, cioè l'uso coordinato dello strumento di ottimizzazione e dello schema di controllo S-MPC comprendente il controllore di loop e il bias di controllo originato dall'ottimizzazione, avendo il meccanismo di attivazione in atto.

Quanto all'utilizzo della stanza si pensa a due scenari:

- un ufficio, occupato appunto in orario d'ufficio e in cui successivamente entra del personale per le pulizie,
- e un'abitazione privata in giornata lavorativa, in cui le persone sono presenti dalla sera alla mattina successiva.

I modelli utilizzati per l'ottimizzazione e per il controllo (simulato) sono rappresentati come diagrammi Modelica nelle Figure 6.1 e 6.2, rispettivamente.

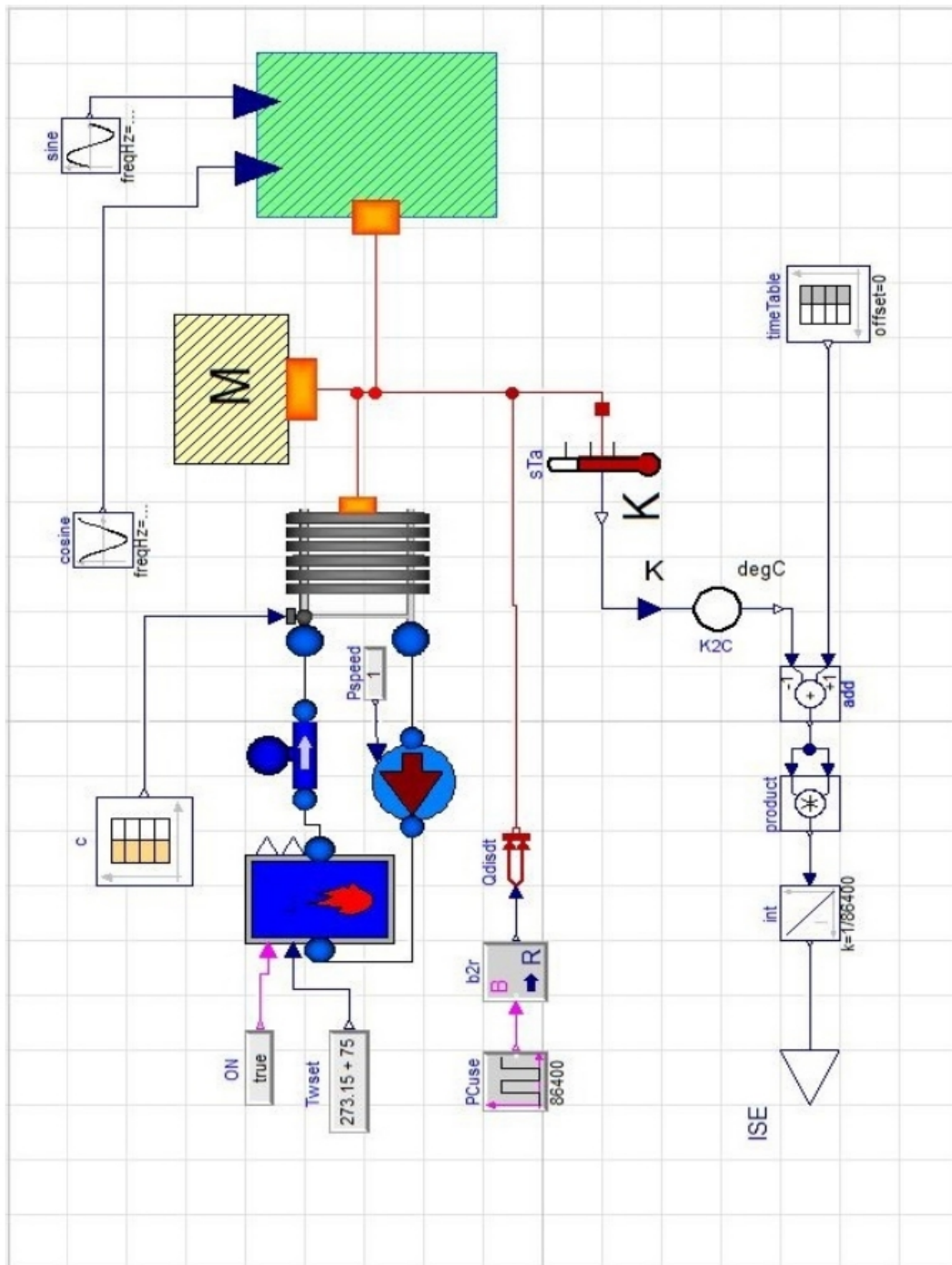


Figura 6.1 : Modello per l'ottimizzazione (MO) nell'esempio di simulazione presentato.

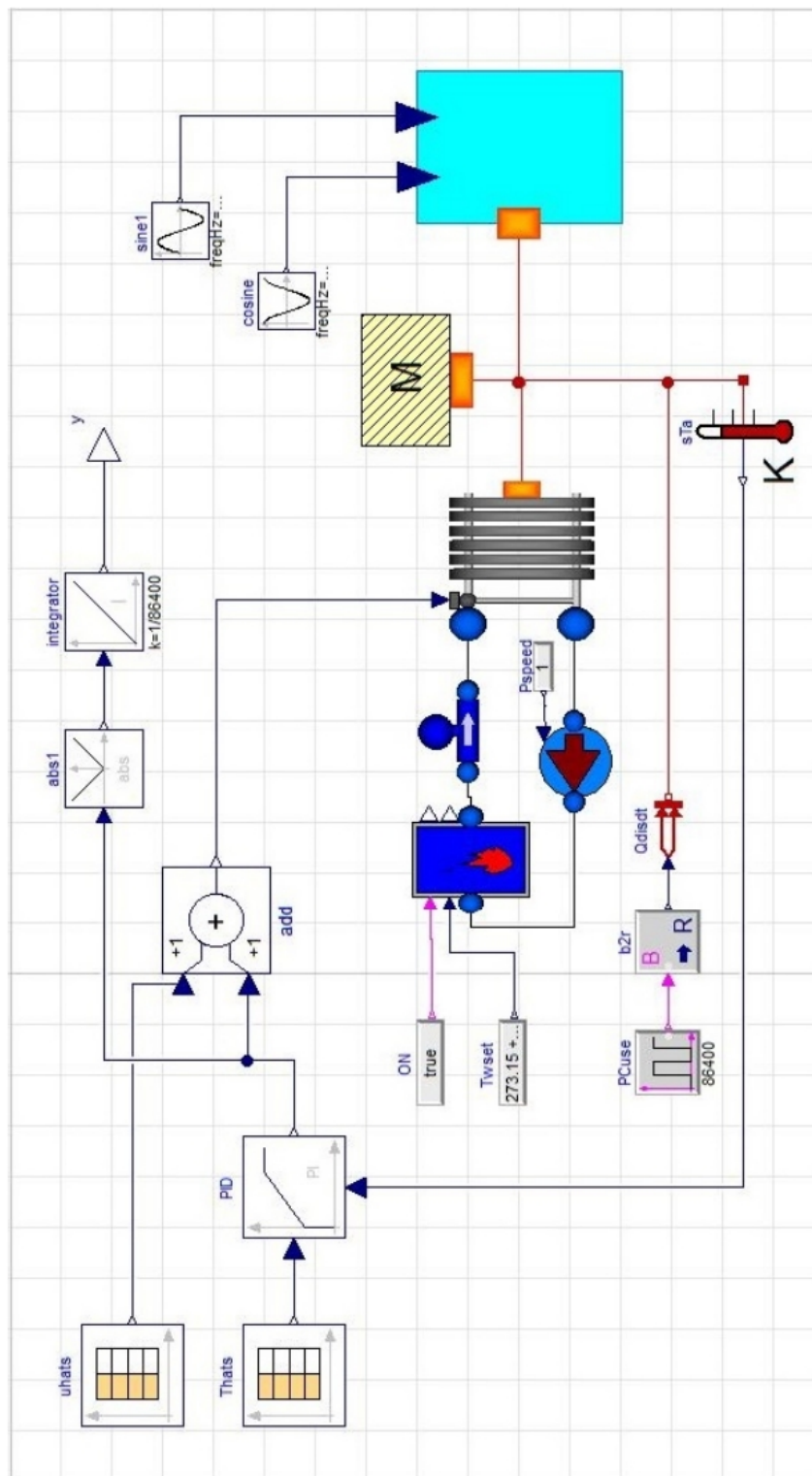


Figura 6.2 : Modello per il controllo (MC) nell'esempio di simulazione presentato.

Per esplorare una certa varietà di condizioni operative, sono stati testati due profili di set point e due condizioni ambientali esterne. In maggior dettaglio rispetto a quanto anticipato, le scelte fatte sono le seguenti.

- Il primo profilo di set point (d'ora in poi SP1) è tipico ad esempio di un edificio per uffici: fornisce una temperatura adeguata durante l'orario di lavoro, quando le persone non stanno svolgendo un'attività fisica significativa, poi abbassa la temperatura quando gli occupanti sono essenzialmente il personale addetto a pulizie e manutenzioni, che al contrario esercita uno sforzo fisico e può non apprezzare un clima troppo caldo, quindi abbassa ulteriormente la temperatura per la notte.
- Il secondo profilo di set point (SP2) può essere quello di un'abitazione durante una giornata lavorativa, quando gli occupanti sono presenti la mattina e nel tardo pomeriggio/sera; la temperatura desiderata assume quindi due valori, il più caldo in presenza degli occupanti durante il giorno, il più fresco in loro assenza e durante la notte.
- La prima condizione esterna (EC1) si riferisce ad una giornata invernale ma presenta un'escursione giornaliera significativa, rappresentando così ad esempio il clima di un comune a latitudini medie.
- La seconda condizione esterna (EC2) è più dura in quanto comprende costantemente una temperatura fredda, come potrebbe essere il caso ad esempio in climi estremi.

Vi è quindi una certa variabilità tra gli esperimenti simulati e, sebbene questo sia solo un test preliminare dell'applicazione completa S-MPC, l'ottimizzatore deve affrontare una certa diversità di situazioni. L'ottimizzazione viene eseguita ad ogni ora del tempo simulato, mentre il controllore di loop è rappresentato per semplicità ed efficienza nel tempo continuo.

Nel seguito sono riportati i risultati di alcune delle prove di simulazione eseguite coi modelli, le condizioni e la modalità descritte sopra.

6.1 Prova 1

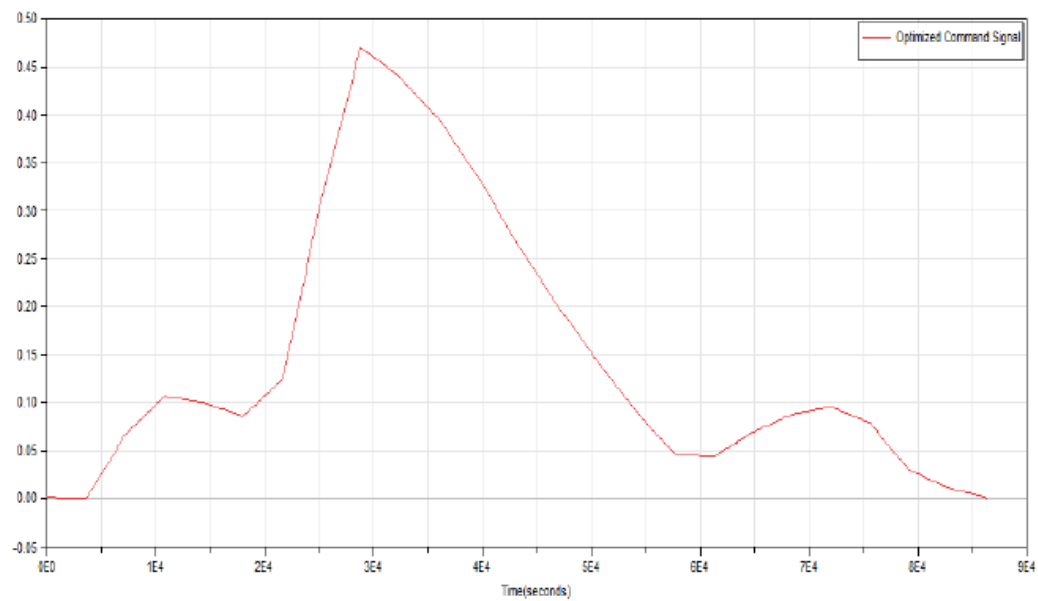


Figura 6.3 : Prova 1 - Segnale di comando ottimizzato per il riscaldatore.

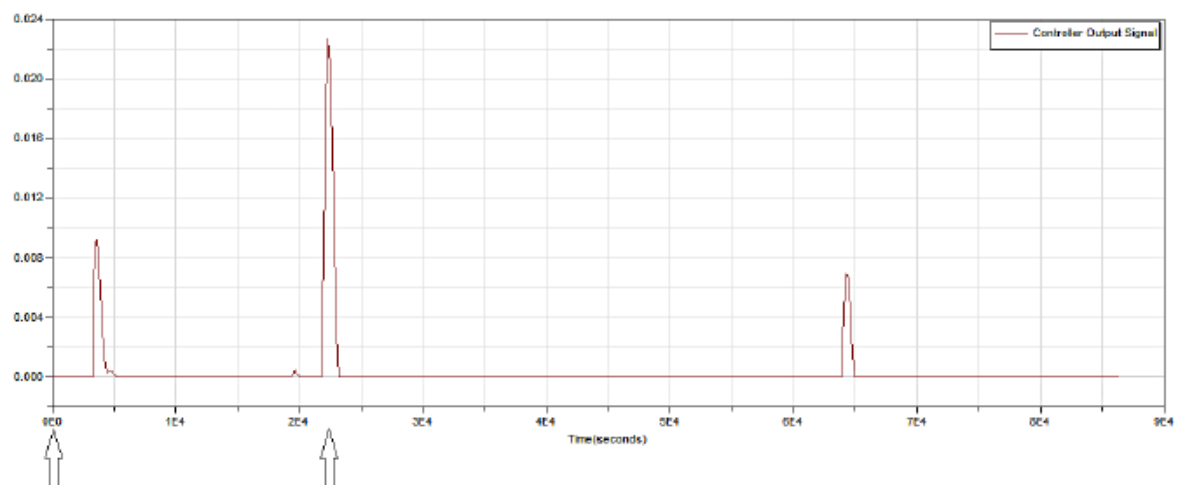


Figura 6.4 : Prova 1 - Uscita del controllore del loop.

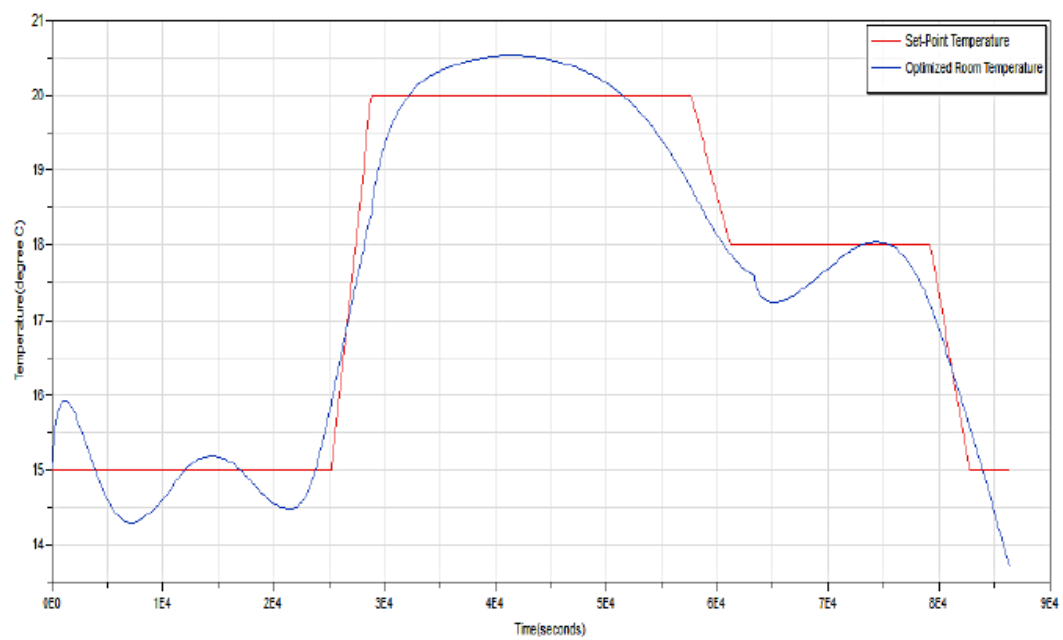


Figura 6.5 : Prova 1 - Set point e temperatura ottimizzata.

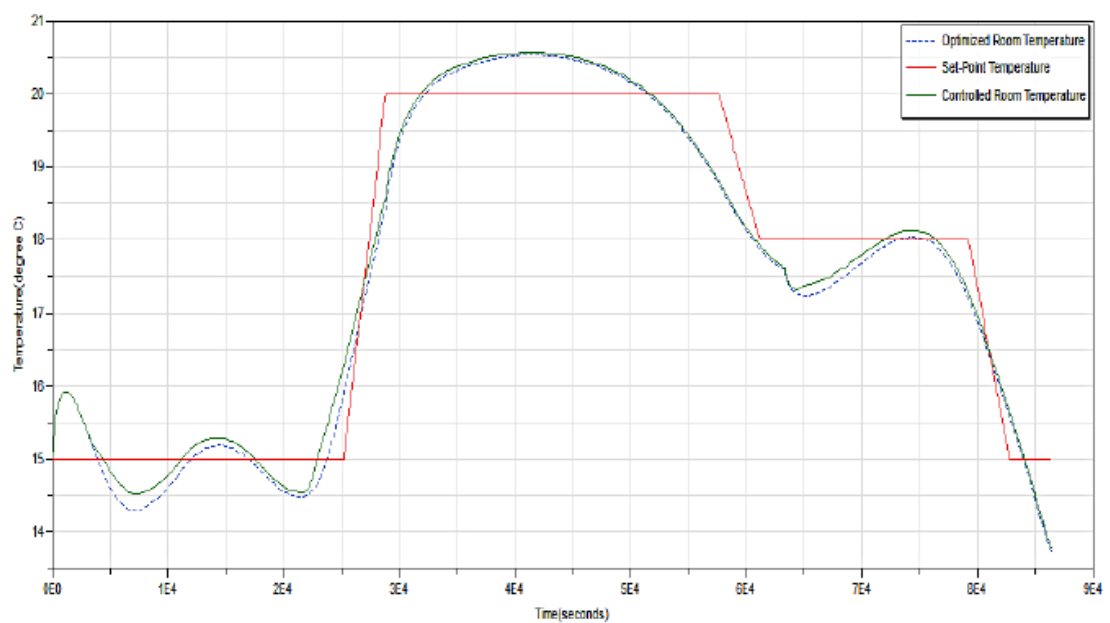


Figura 6.6 : Prova 1 - Set point ottimizzato e temperatura controllata.

Questa prova si riferisce a SP1 e EC1 e la presentazione dei risultati è organizzata come segue.

- La Figura 6.3 mostra il risultato delle ottimizzazioni, ovvero il bias per il controllore di loop. Il valore minimo per questo segnale - che è il comando del riscaldatore ambiente - è 0, mentre il suo massimo è 1.
- La Figura 6.4 illustra l'uscita del controllore di loop.
- La Figura 6.5 mostra quale sarebbe il comportamento della temperatura controllata in assenza del controllore di loop.
- La Figura 6.6, infine, riporta il risultato dell'intero schema, ovvero il comportamento della temperatura del controllore come risultante dall'azione combinata di ottimizzatore e controllore di loop.

Come mostra la Figura 6.3 all'inizio, il riscaldatore doveva essere azionato con una bassa intensità di circa 0,1 (poiché l'ufficio è chiuso) fino all'inizio dell'orario di apertura dell'ufficio, quindi l'intensità è aumentata fino a 0,46 in modo che il riscaldatore potesse fornire abbastanza calore per raggiungere il set point. Man mano che l'orario di ufficio sta per finire, il segnale di comando diminuisce drasticamente fino a circa 0,1 e alla fine terminando anche i servizi di pulizia arriva a 0. La Figura 6.4 illustra l'output del controllore. Il valore massimo 0,023 dell'uscita è già stato definito come soglia che funge da trigger di ottimizzazione, quando il valore dell'uscita del controllore supera questo valore il sistema attiverà nuovamente l'ottimizzazione e questa continuerà fino a quando l'uscita del controllore rimane al di sotto la soglia. In questa prova il problema di ottimizzazione doveva essere risolto solo due volte, indicate dalle frecce.

La Figura 6.5 mostra i soli valori ottimizzati del segnale di comando (in base al set point dato ma senza l'ausilio del controllore di loop).

La Figura 6.6 mostra il valore della temperatura ambiente con l'intero schema S-MPC. Come si può vedere il controllore di loop interviene, il che è sensato, ma non tutti i suoi interventi si traducono in una ri-ottimizzazione, il che conferma l'utilità dello schema.

6.2 Prova 2

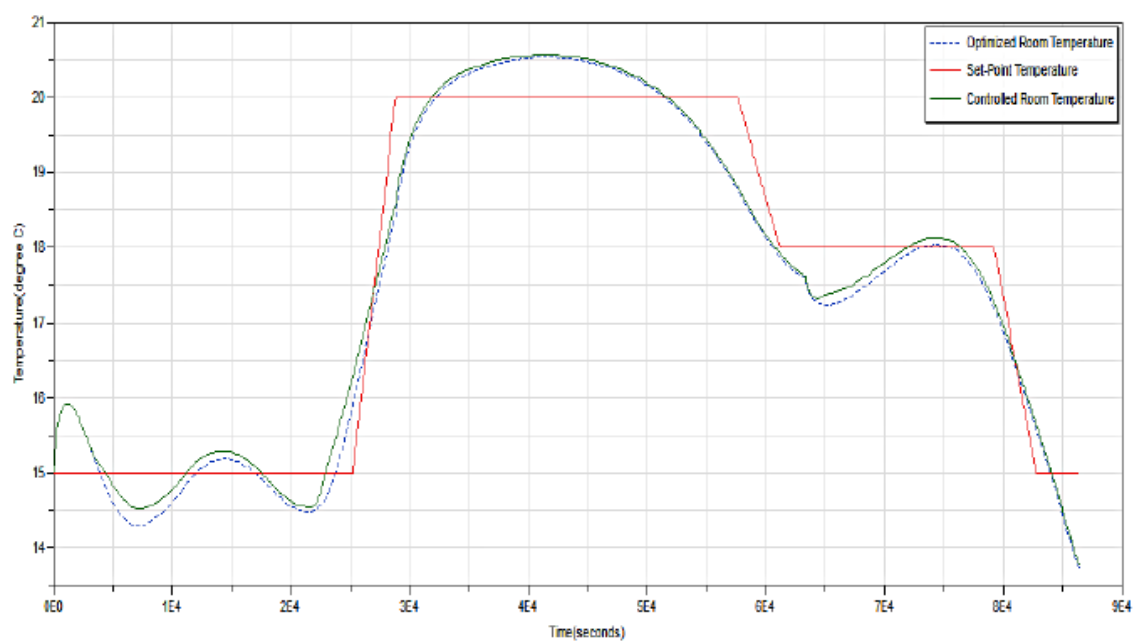


Figura 6.7 : Prova 2 - Segnale di comando ottimizzato per il riscaldatore.

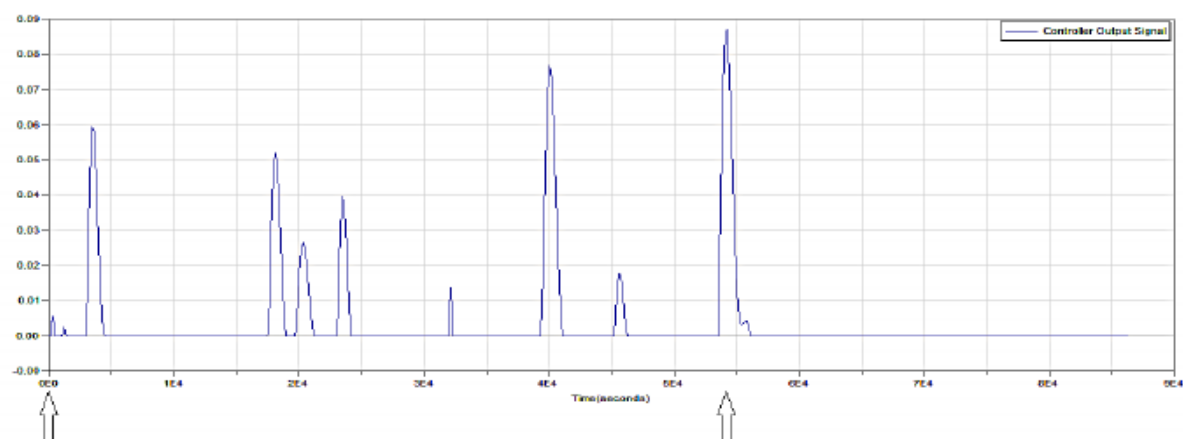


Figura 6.8 : Prova 2 - Uscita del controllore del loop.

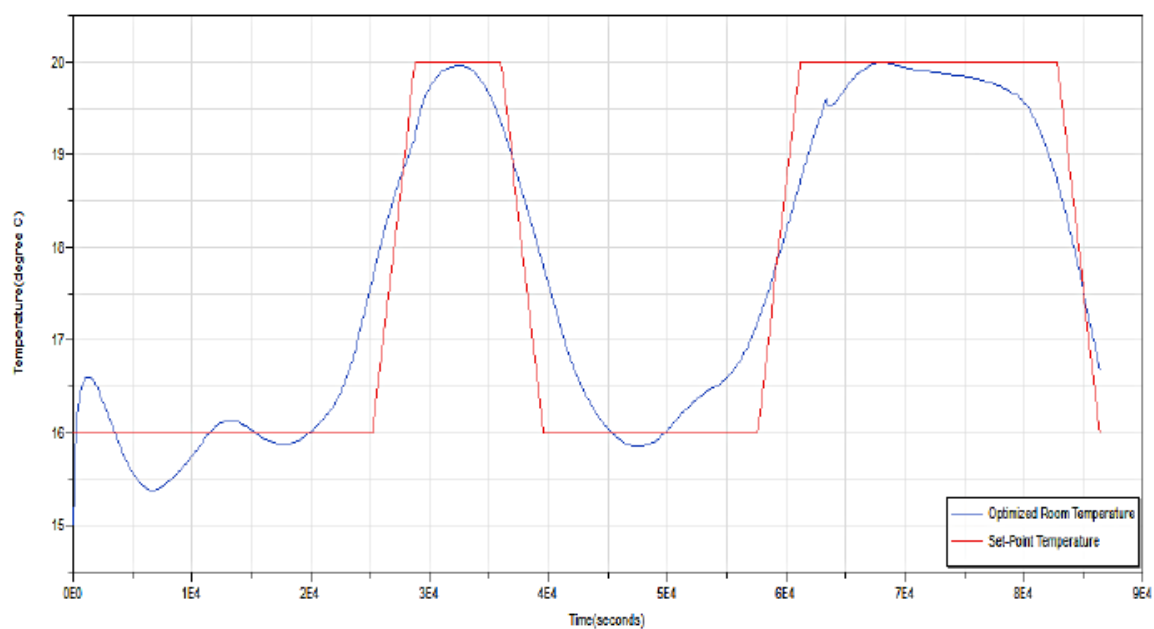


Figura 6.9 : Prova 2 - Set point e temperatura ottimizzata.

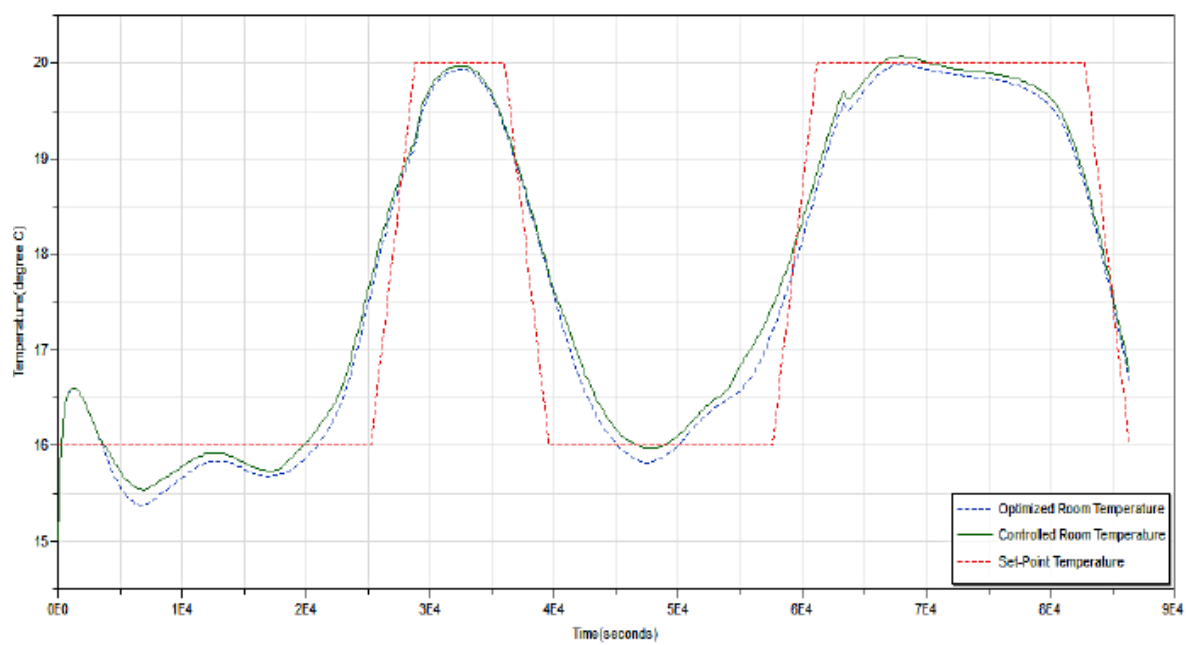


Figura 6.10 : Prova 2 - Set point ottimizzato e temperatura controllata.

Questa prova si riferisce a SP2 e EC1. I risultati sono nelle Figure da 6.7 a 6.10, che sono organizzate allo stesso modo delle Figure da 6.3 a 6.6.

Poiché la giornata inizia con una temperatura fredda, in quel periodo è necessario uno sforzo maggiore per aumentare la temperatura. Dopo le 23:00, al contrario, solo valori piccoli per il segnale di comando possono fornire sufficiente calore per le ore successive.

Anche qui le ottimizzazioni eseguite sono indicate con frecce e valgono le stesse considerazioni fatte per la prova 1.

6.3 Prova 3

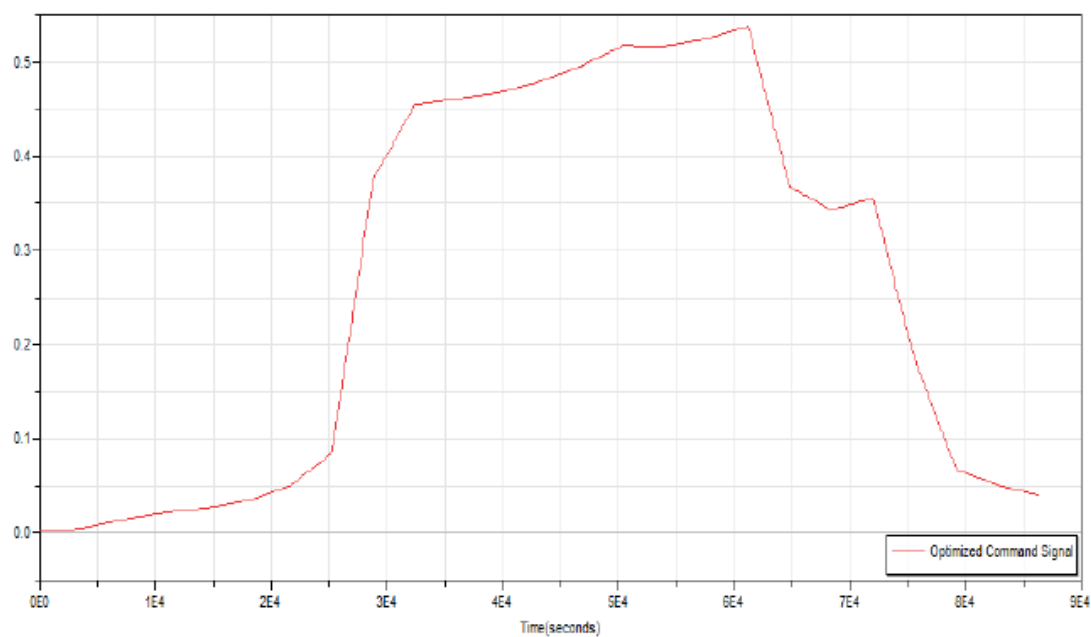


Figura 6.11 : Prova 3 - Segnale di comando ottimizzato per il riscaldatore.

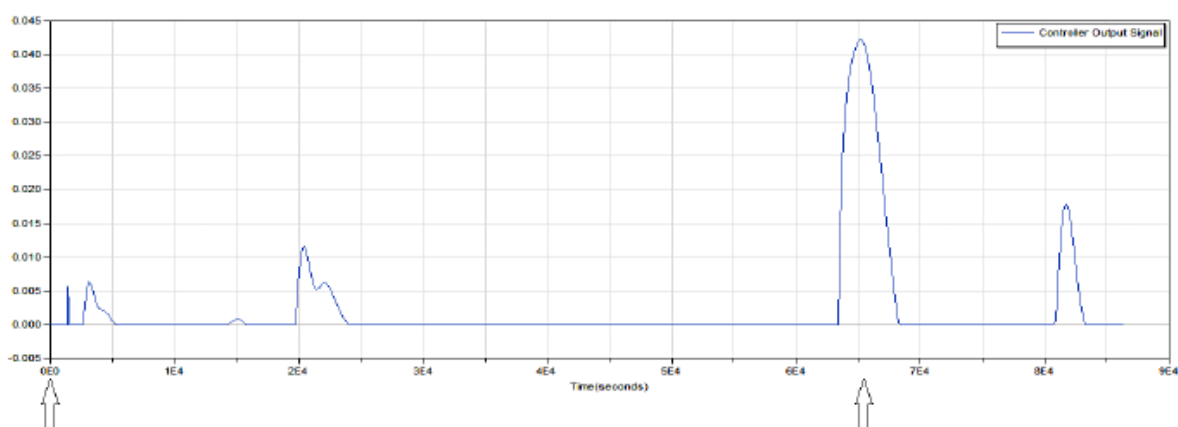


Figura 6.12 : Prova 3 - Uscita del controllore del loop.

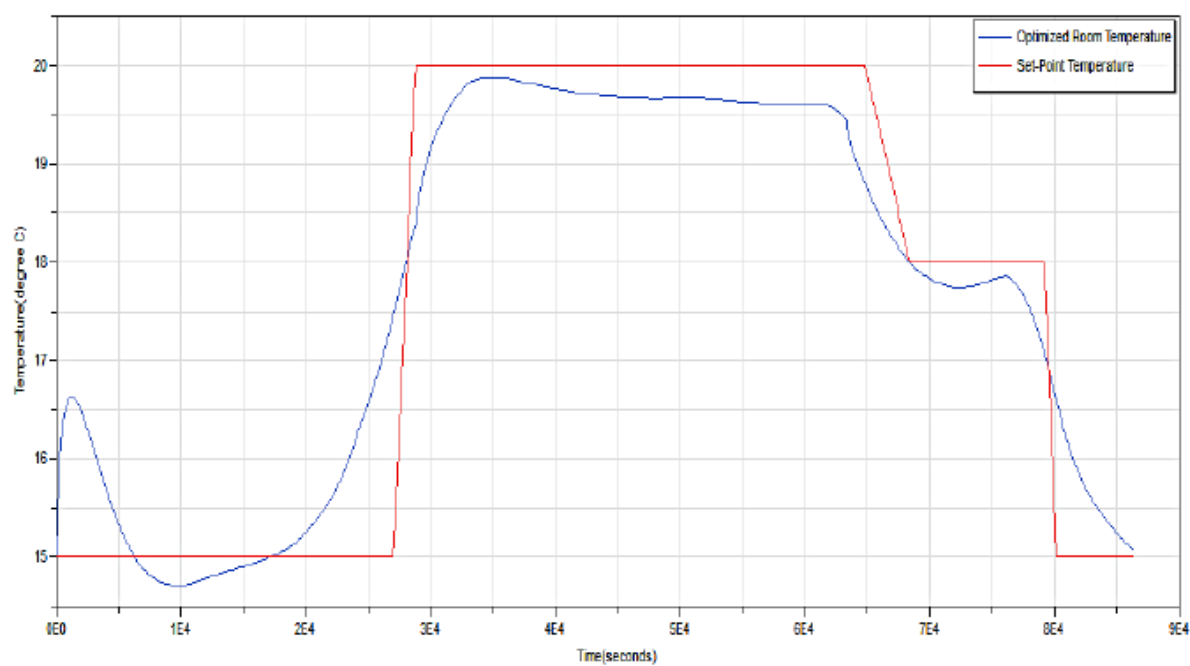


Figura 6.13 : Prova 3 - Set point e temperatura ottimizzata.

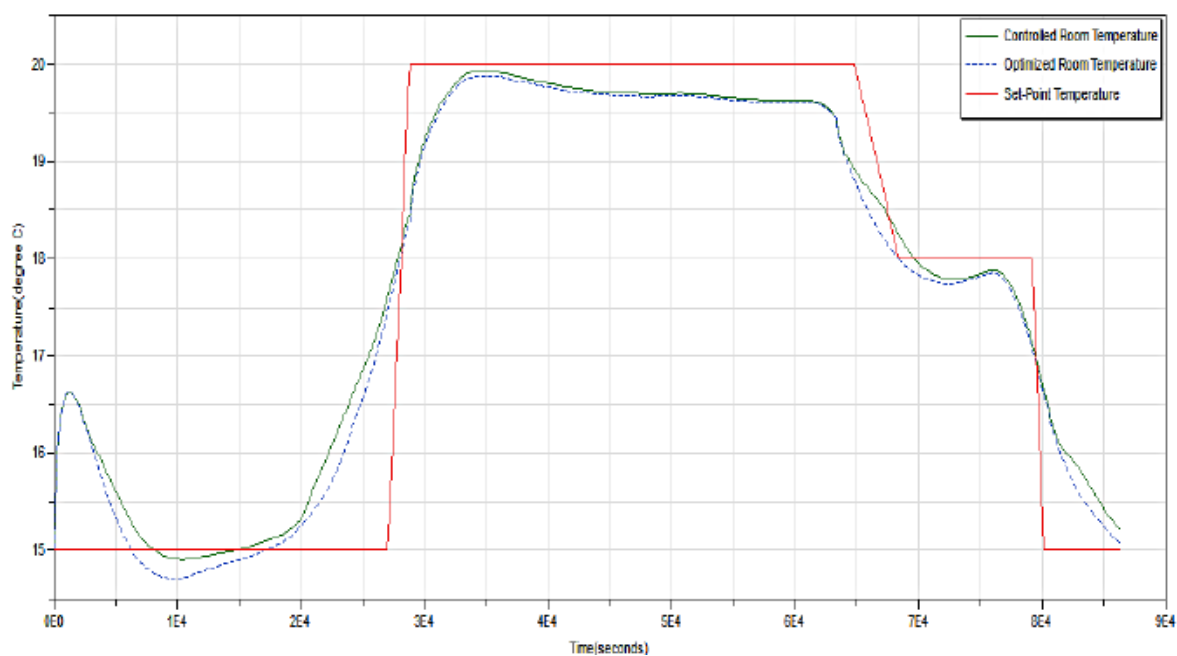


Figura 6.14 : Prova 3 - Set point ottimizzato e temperatura controllata.

Questa prova si riferisce a SP1 e EC2. I risultati sono nelle Figure da 6.11 a 6.14, organizzate nello stesso modo delle prove precedenti.

Poiché le temperature iniziali per il volume d'aria e le pareti sono di circa 15 gradi, la temperatura della stanza diminuisce drasticamente da quel valore quindi la stanza necessita di meno sforzo nelle prime 8 ore dal riscaldatore perché il sistema di riscaldamento fornisce già la temperatura di set point nel giorno precedente.

Ancora una volta le ottimizzazioni effettuate vengono indicate con delle frecce e valgono le stesse considerazioni fatte prima.

6.4 Prova 4

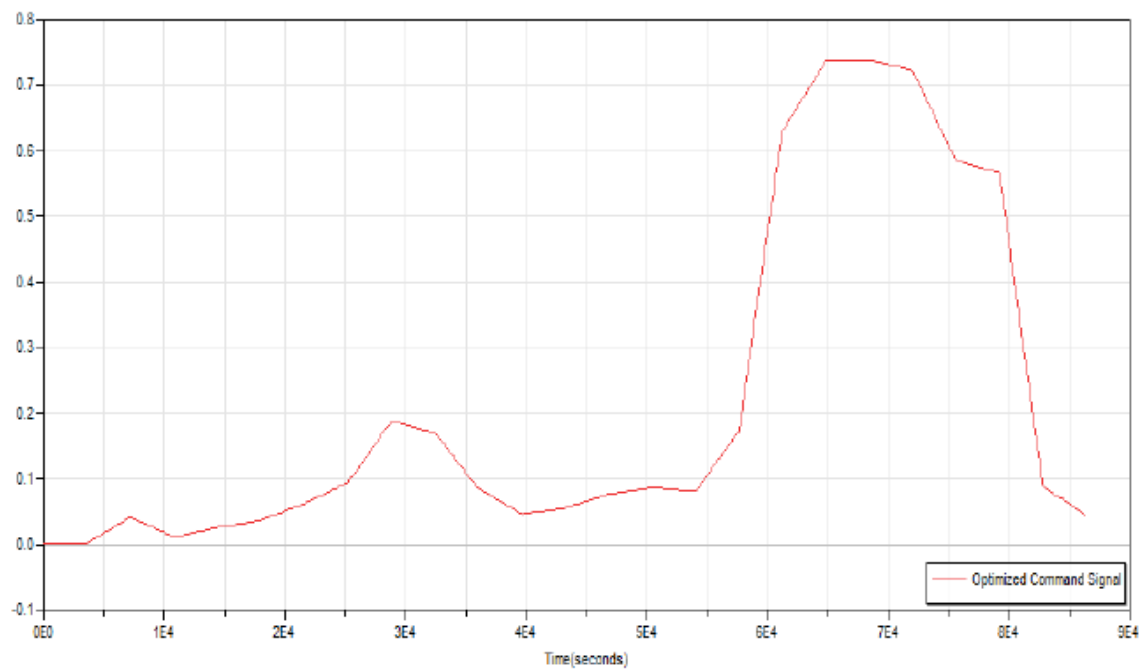


Figura 6.15 : Prova 4 - Segnale di comando ottimizzato per il riscaldatore.

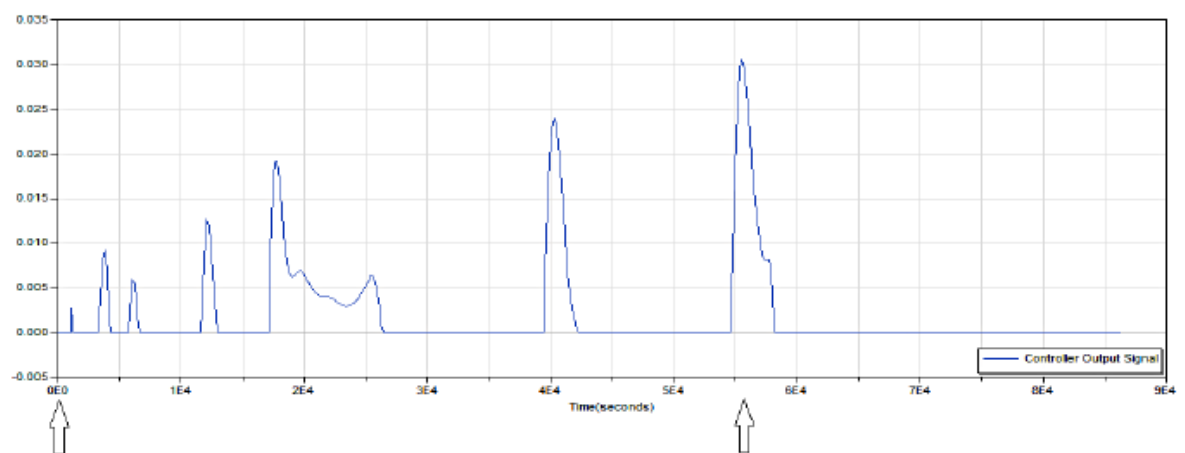


Figura 6.16 : Prova 4 - Uscita del controllore del loop.

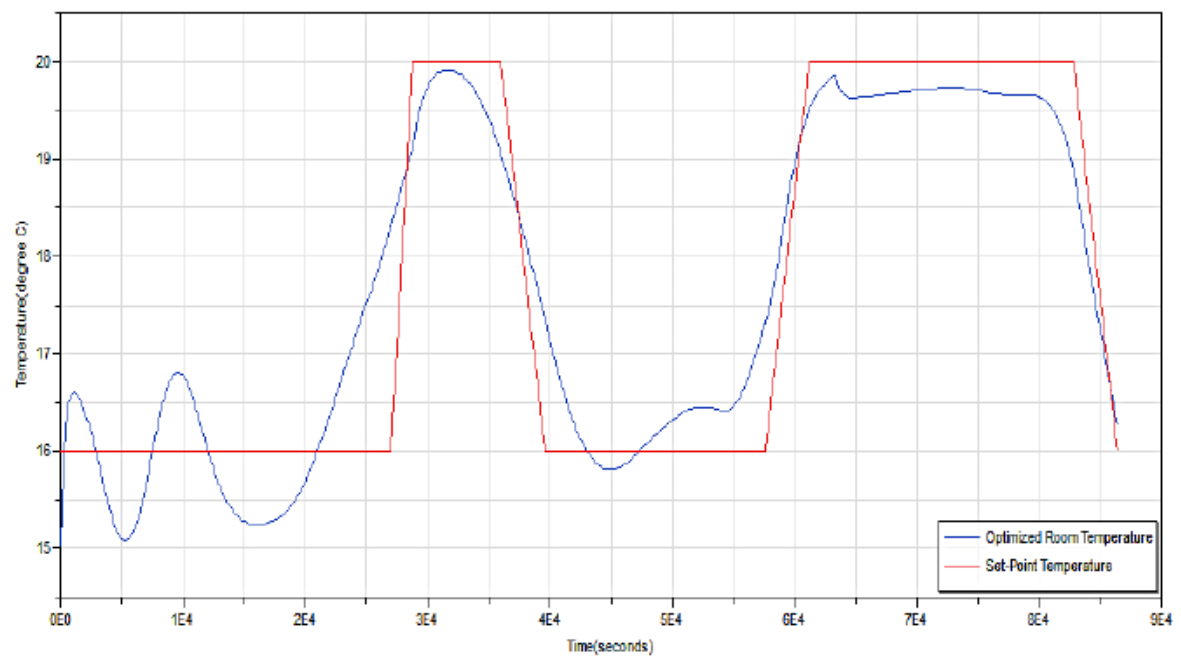


Figura 6.17 : Prova 4 - Set point e temperatura ottimizzata.

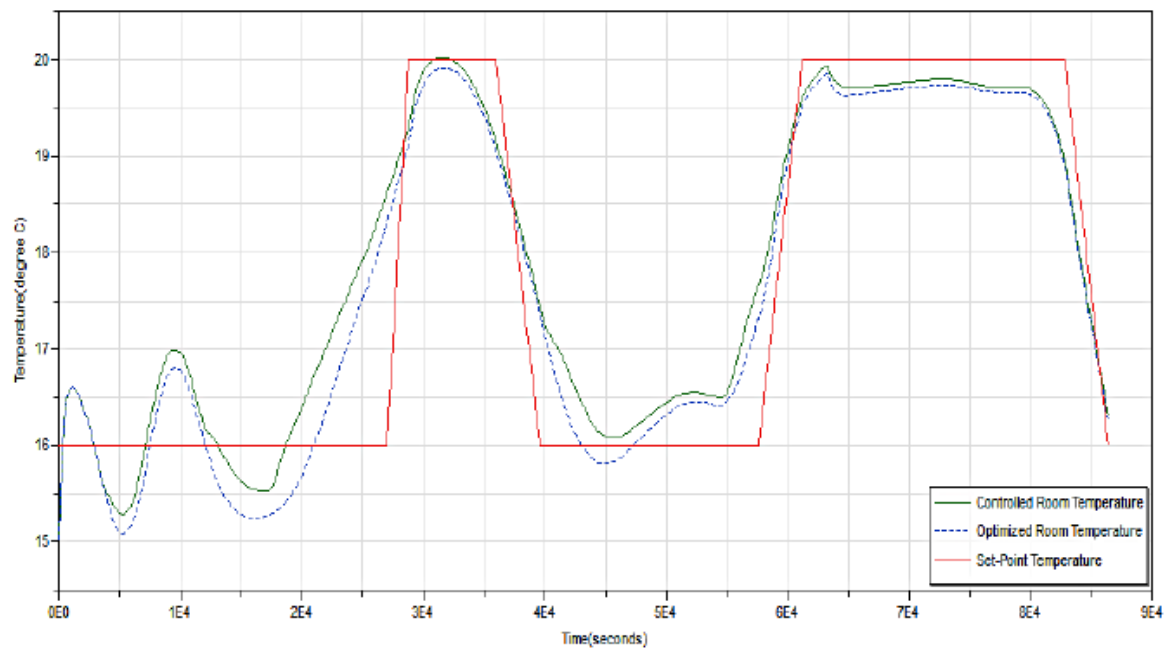


Figura 6.18 : Prova 4 - Set point ottimizzato e temperatura controllata.

Questa prova si riferisce a SP2 e EC2. I risultati sono nelle figure da 6.15 a 6.18, organizzati come sopra.

Qui nelle prime 8 ore il fabbisogno del riscaldatore è molto inferiore rispetto alle ore successive poiché abbiamo ipotizzato che il riscaldatore sia stato utilizzato anche il giorno prima quindi la temperatura della stanza in partenza non è uguale alla temperatura esterna istantaneamente. Un altro motivo di questa differenza è che il set point che deve essere di 20 gradi per 6 ore forza l'accensione continua del riscaldatore ad alta intensità.

Anche in questo caso le frecce indicano le ottimizzazioni e si riscontra di nuovo l'utilità dello schema.

7 Conclusioni e sviluppi futuri

La tecnica Sporadic Model-based Predictive Control (S-MPC) introdotta di recente è stata qui resa parte di un ambiente di modellazione e simulazione basato sul linguaggio Modelica e utilizzando JModelica come strumento di ottimizzazione.

Per migliorare ulteriormente i risultati, sono stati creati e testati modelli Modelica che sfruttano esplicitamente i controlli periferici installati nella maggior parte dei sistemi energetici degli edifici. Ciò ha permesso di ottenere una facile parametrizzazione e un'elevata efficienza computazionale.

È stata proposta un'applicazione python per racchiudere tutti i componenti realizzati in un unico ambiente, che è stato testato con risultati incoraggianti.

Il lavoro futuro perseguirà un'ulteriore ingegnerizzazione della presente applicazione. Inoltre, verranno studiati vari punti - ad esempio, la scelta e la messa a punto del controllore di loop o una migliore calibrazione del trigger di ottimizzazione - per i quali le soluzioni formali sono estremamente macchinose da derivare e molto probabilmente di scarsa (e forse quasi nulla) utilità pratica al confronto di estese campagne di simulazione condotte come quella qui presentata, dalle quali possono facilmente scaturire le opportune procedure operative.

Riferimenti bibliografici

- [1] Y. Ma, F. Borrelli, B. Hancey, A. Packard and S. Bortoff. "Model predictive control of thermal energy storage in building cooling systems. in: Proc. of the 48th IEEE conference on decision and control.
- [2] ISO 7730 (international standard). moderate thermal environments—determination of the PMV and PPD indices and specification of the conditions for thermal comfort, 1994.
- [3] F. Allard, V. Dorer and H. Feustel. Fundamentals of the multizone airflow model; COMIS-Technical Note 29. Air Infiltration and Ventilation Centre., Coventry, UK., 1990.
- [4] ASHRAE. Standard for Ventilation and acceptable indoor air quality in low-rise residential buildings, volume 62. ASHRAE, 2003.
- [5] ASHRAE Handbook-Fundamentals. ASHRAE, 2005.
- [6] M. Bonvini and A. Leva. Object-oriented sub-zonal modelling for efficient energy-related building simulation. *Mathematical and Computer Modelling of Dynamical Systems: Methods, Tools and Applications in Engineering and Related Sciences*, 17:6:543–559, 2011.
- [7] M. Bonvini, M. Popovac, and A. Leva. Sub-zonal computational fluid dynamics in an object-oriented modelling framework. *Building Simulation*, 7:439–454, 2014.
- [8] P. Campo and M. Morari. Robust model predictive control. *Proc. American Control Conference*, 1987.
- [9] K. Chen, Y. Jiao, and E. S. Lee. Fuzzy adaptive networks in thermal comfort. *Applied Mathematics Letters*, 19(5):420–426, 2006.
- [10] Q. Chen. Ventilation performance prediction for buildings: A method overview and recent applications. *Building and Environment*, 44:848–858, 2009.
- [11] Q. Chen and W. Xu. A zero equation turbulence model for indoor airflow simulation. *Energy and Buildings*, 28:137–144, 1998.
- [12] T. Chen. Application of adaptive predictive control to a floor heating system with a large thermal lag. *Energy and Buildings*, 34(1):45–51, (2002).
- [13] S. Cho and M. Zaheer-uddin. Predictive control of intermittently operated radiant floor heating systems. *Energy and Conversion Management*, 44:1333–1342, 2012.
- [14] J. Cigler. Model Predictive Control for Buildings. PhD thesis, Faculty of Electrical Engineering, Prague, June 2013.

- [15] E. Djunaedy, J. Hensen, and M. Loomans. Towards external coupling of building energy and air flow modeling programs. *ASHRAE Trans.*, 109(2):771–787, 2003.
- [16] A. Dounis and C. Caraiscos. Advanced control systems engineering for energy and comfort management in a building environment—a review. *Renewable and Sustainable Energy Reviews*, 13:1246–1261, 2009.
- [17] P. O. Fanger. *Thermal comfort: analysis and applications in environmental engineering*. McGraw-Hill, New York, 1972.
- [18] G. Follen, C. Kin, I. Lopez, J. Sang, and S. Townsend. A CORBA based development environment for wrapping and coupling legacy scientific codes. 10th IEEE International Symposium on High Performance Distributed Computing, San Francisco, CA, IEEE, New York, page 22–31, 2001.
- [19] R. Freire, G. Oliveira, and N. Mendes. Predictive controllers for thermal comfort optimization and energy savings. *Energy and Buildings*, 40(7):1353–1365, 2008.
- [20] M. Gwerder, J. Tödtli, B. Lehmann, F. Renggli, and V. Dorer. Control of thermally activated building systems. In *Proceedings: 9th REHVA World Congress Clima*, 2007.
- [21] P. Haves, T. Salsbury, C. Claridge, and M. Liu. Use of whole building simulation in online performance assessment: Modeling and implementation issues. *Proceedings of the 7th International IBPSA Conference, Rio de Janeiro, Brazil, International Building Performance Simulation Association (IBPSA)*, page 335–342, 2001.
- [22] I. Hazyuk, C. Ghiaus, and D. Penhouet. Optimal temperature control of intermittently heated buildings using model predictive control: Part ii - control algorithm. *Building and Environment*, 51:388–394, 2012.
- [23] A. Leva and G. Magnani. Detecting process variations in low-end PID autotuners. In *Proc. IFAC World Congress b’02, Barcelona, Spain*, 2002.
- [24] N. Li, Z. Wang, W. Liu, X. Peng, and G. Gong. Research on construction and interoperability of complex distributed simulation system. *Lect. Notes Comput. Sci.* 3398, page 131–140, 2005.
- [25] Z. Liao and A. Dexter. An inferential model-based predictive control scheme for optimizing the operation of boilers in building spaceheating systems. *IEEE Transactions on Control Systems Technology*, 18(5):1092–1102, (2010).
- [26] J. Maciejowski. *Predictive Control with Constraints*. Prentice Hall, 2002.
- [27] A. Leva, F.M. Benzi, V. Magagnotti, and G. Vismara. Sporadic model predictive control (paper in preparation).

- [28] L. Mora, A. Gadgil, and E. Wurtz. Comparing zonal and cfd model predictions of isothermal indoor airflows to experimental data. *Indoor Air*, 23:77–85, 2003.
- [29] P. Morosan, R. Bourdais, D. Dumur, and J. Buisson. Building temperature regulation using a distributed model predictive control. *Energy and Buildings*, 42(9):1445–1452, 2010.
- [30] T. R. of the University of California (through Lawrence Berkeley National Laboratory). Description of GenOpt.
- [31] T. B. of Trustees of the University of Illinois and the Regents of the University of California through the Ernest Orlando Lawrence Berkeley National Laboratory. Description of EnergyPlus.
- [32] F. Oldewurtel, A. Parisio, C. N. Jones, D. Gyalistras, M. Gwerder, V. Stauch, B. Lehmann, and M. Morari. Use of model predictive control and weather forecasts for energy efficient building climate control. *Energy and Buildings*, 45:15–27, 2012.
- [33] S. Privara, J. Siroky, L. Ferkl, and J. Cigler. Model predictive control of a building heating system: The first experience. *Energy and Buildings*, 43:564–572, 2011.
- [34] J. Rawlings and J. Muske. The stability of constrained receding horizon control. *IEEE Transaction on Automatic Control*, 38(10):1512–1516, 1993.
- [35] Z. Ren and J. Stewart. Simulating air flow and temperature distribution inside buildings using a modified version of comis with sub-zonal divisions. *Energy and Buildings*, 35:257–271, 2003.
- [36] J. Richalet, A. Rault, J. Testud, and J. Papon. Algorithmic control of industrial process, in: *Proceedings: Symposium on Identification and System Parameter Estimation*. IFAC, Tbilisi, 1976.
- [37] P. Sahlin. The methods of 2020 for building envelope and HVAC systems simulation – will the present tools survive? In *Proceedings of the CIBSE Conference*, Dublin, Ireland, 2000.
- [38] J. Siroky, F. Oldewurtel, J. Cigler, and S. Privara. Experimental analysis of model predictive control for an energy efficient building heating system. *Applied Energy*, 88 (9):3079–3087, 2011.
- [39] S. Skogestad. Simple analytic rules for model reduction and PID controller tuning. *Journal of process control*, 13(4):291–309, 2003.
- [40] S. Taylor, A. Bruzzone, R. Fujimoto, B. Gan, S. Strassburger, and R. Paul. Distributed simulation and industry: Potentials and pitfalls. In *Proceedings of the 2002 Winter Simulation Conference*, 2002.
- [41] M. Wetter. Modelica-based modeling and simulation to support research and development in building energy and control systems. *J. Build. Perform. Simul.*, 2 (1):143–161, 2009.

- [42] M. Wetter and P. Haves. A modular building controls virtual test bed for the integration of heterogeneous systems. In Proceedings of the 3rd National Conference of IBPSA-USA, Berkeley, CA, 2008.
- [43] A. Zheng and M. Morari. Stability of model predictive control with soft constraints. Internal Report. Californian Institute of Technology, 1994.
- [44] W. Zuo and Q. Chen. Fast and informative flow simulations in a building by using fast fluid dynamics model on graphics processing unit. *Building and Environment*, 45:747–757, 2010.
- [45] W. Zuo, J. Hu, and Q. Chen. Improvements in ffd modeling by using different numerical schemes. *Numerical Heat Transfer, Part B: Fundamentals*, 58:1–16, 2010.