Politecnico di Milano

Facoltà di Ingegneria

Dipartimento di Elettronica e Informazione

Corso di Laurea Specialistica in

Ingegneria per l'Ambiente ed il Territorio

Dynamic emulation modelling per la gestione di un serbatoio a rilascio selettivo

Relatore:
Prof. Rodolfo Soncini-Sessa

Correlatore:
Ing. Stefano Galelli

Tesi di Laurea di:
Flavio Rossini

Matr. **725173**

Anno Accademico **2009/2010**

Ringraziamenti

Numerose sono le persone che devo ringraziare, e troppo esiguo lo spazio per poterlo fare. Ringrazio sentitamente tutti coloro che hanno reso possibile questo lavoro: in primo luogo il professor Soncini-Sessa per l'opportunità concessami; Stefano Galelli per avermi seguito scrupolosamente dall'inizio alla fine del presente lavoro, dimostrando peraltro una notevole pazienza; Andrea Castelletti ed Enrico Weber, il consiglio dei quali è sempre stato prezioso; Marcello Restelli, che mi ha aiutato a risolvere i numerosi problemi tecnici e informatici che hanno costellato il lavoro. Ringrazio inoltre tutte le persone, tesisti e dottorandi, che hanno supportato il mio lavoro con utili suggerimenti, in particolare Matteo e Paolo. Un ringraziamento speciale va a Tiuz, Grana e Fede che hanno condiviso con me questa avventura fin dagli inizi, e che hanno contribuito a rendere questi anni di studio anche divertenti. Infine, ma non per importanza, ringrazio la mia famiglia per il sostegno che non mi è mai mancato, soprattutto nei momenti di maggiore difficoltà.

Indice

Introduzione

Nell'ambito della modellizzazione di grandi sistemi ambientali, i progressi della conoscenza scientifica e l'aumento della potenza dei calcolatori rendono possibile l'adozione di modelli matematici di tipo *process-based*. La complessità delle strutture di questi modelli pone però forti limitazioni in termini di implementazione pratica, in particolare per i problemi di controllo ottimo, come ad esempio il progetto delle politiche di regolazione di laghi e serbatoi. Tale limite può essere superato con tecniche di *emulation modelling*, che consistono nella costruzione di un modello ridotto e computazionalmente efficiente che possa sostituire un modello *process-based* nella risoluzione di problemi particolarmente onerosi dal punto di vista computazionale. Nel presente lavoro, si affronta questo problema adottando una procedura (Galelli, 2010) per l'identificazione di un *emulation model*, allo scopo di progettare politiche di gestione applicate ad un serbatoio munito di un sistema di rilascio selettivo (SWS).
Per identificare l'emulation model, viene utilizzato un set di dati generati con un modello distribuito *process-based*. L'emulation model viene poi usato per progettare una politica di gestione giornaliera attraverso un algoritmo (*fitted Q-iteration*) basato su regressori di tipo tree-based. Questo approccio sarà testato sul serbatoio di Tono, un bacino artificiale giapponese utilizzato per diversi scopi, tra cui la fornitura idrica per l'irrigazione e la produzione idroelettrica, ed in cui sono rilevanti alcuni problemi di qualità idrica (torbidità e fioriture algali).

Il presente lavoro è strutturato come segue. Il primo capitolo fornisce una breve panoramica delle tecniche disponibili per la riduzione di modelli *process-based*, mentre il secondo capitolo descrive nel dettaglio la procedura utilizzata per l'identificazione del modello ridotto. Il terzo capitolo descrive le caratteristiche del sistema di Tono, individua i portatori d'interesse ed i corrispondenti indicatori, e introduce le possibili strategie per la risoluzione del problema di controllo ottimo. Nel quarto

capitolo, viene applicata la procedura di identificazione dell'emulation model al caso in esame, e ne vengono esposti i risultati. Infine, nel quinto ed ultimo capitolo, vengono mostrate le prestazioni delle politiche di controllo progettate: tali risultati sono confrontati con quelli relativi alle politiche di controllo progettate basandosi sulle indicazioni di un approccio empirico *expert-based* sviluppato nell'ambito di lavori precedenti.

Capitolo 1

Strategie per la riduzione di modelli process-based

I progressi della conoscenza scientifica e l'aumento della potenza dei calcolatori hanno fortemente migliorato il livello di conoscenza necessario per la costruzione di modelli Process-Based (PB), che vengono ampiamente utilizzati per la modellizzazione di grandi sistemi ambientali. La maggiore complessità delle strutture di tale modello pone però forti limitazioni in termini di implementazione pratica, in particolare per i problemi che tipicamente richiedono centinaia se non migliaia di *model evaluations*, come, ad esempio, l'analisi di sensitività, l'analisi di scenario e il controllo ottimo. Recentemente, un'attenzione sempre maggiore viene riservata all'*emulation modelling* come modalità per superare tale limite. Un *emulation model* è un modello semplificato, e computazionalmente efficiente, di un modello process-based, che può essere usato per risolvere problemi particolarmente esigenti in termini di risorse di calcolo. Un modello di questo tipo può essere derivato semplificando la struttura del modello process-based, o identificato sulla base dei dati prodotti da quest'ultimo tramite simulazione.
Poiché il numero dei problemi che possono trarre beneficio dall'identificazione e dal successivo utilizzo di un emulation model sono molto vasti, così come la quantità di tecniche disponibili a tale scopo, questo paragrafo si occupa dell'analisi e della classificazione di tutti questi problemi, e delle diverse strategie di riduzione.

1.1 Definizione del problema

1.1.1 Il sistema $\mathcal{S}$

Si consideri un sistema ambientale $\mathcal{S}$, affetto da una forzante esogena $\mathcal{W}_t$, possibilmente distribuita nello spazio, e siano lo stato $\mathcal{X}_t$ e l'output $\mathcal{Y}_t$ variabili nel dominio spazio-tempo $\mathcal{L}$ x $\mathcal{T}$. Le applicazioni ingegneristiche sono spesso caratterizzate dall'obiettivo di gestire la dinamica del sistema $\mathcal{S}$ tramite una serie di decisioni, ripetute per l'intera vita del sistema, (ad esempio, decidendo giornalmente la quantità d'acqua da rilasciare da un serbatoio). In questo caso sia $\mathcal{X}_t$ che $\mathcal{Y}_t$ sono influenzati anche da una variabile di controllo $\mathbf{u}_t$, applicata in istanti di tempo discreti, in accordo con il passo temporale di decisione Δt. In alternativa, il sistema $\mathcal{S}$ può non essere influenzato da decisioni antropiche tempo-varianti: in questo caso, la sua dinamica viene influenzata solamente da un cambiamento in una delle caratteristiche del sistema (ad esempio, un cambiamento nel processo di afflusso ad un serbatoio) o da una singola decisione antropica $\mathbf{u}^p$, detta variabile o decisione di pianificazione, assunta costante per l'intera vita del sistema (ad esempio, la costruzione di una diga).

1.1.2 Il modello $\mathcal{M}$

L'approccio scientifico tipicamente adottato nella mmodellizzazione dei sistemi ambientali suggerisce di sfruttare la conoscenza fisica del sistema $\mathcal{S}$ per costruire modelli PB (vedi, ad esempio, Beven (1989), Imberger and Patterson (1989), Blumberg and Mellor (1987) per alcuni articoli relativi alle risorse idriche nell'idrologia, limnologia e oceanografia). I modelli PB possono essere distinti in due famiglie.

Modelli fisicamente basati (tempo-continuo). Il sistema $\mathcal{S}$ viene descritto tramite un modello non lineare, che descrive le dinamiche di $\mathcal{X}_t$ e $\mathcal{Y}_t$ sotto l'effetto di $\mathcal{W}_t$ ed eventualmente $\mathbf{u}_t$. Il modello è costituito da un sistema di equazioni alle derivate parziali, la cui risoluzione numerica richiede la discretizzazione del dominio spazio-tempo $\mathcal{L}$ x $\mathcal{T}$ del sistema $\mathcal{S}$ tramite una griglia di discretizzazione. Il modello può essere rappresentato dalla seguente espressione

$$\dot{\mathcal{X}}_t = f_t(\mathcal{X}_t, \mathcal{W}_t, \mathbf{u}_t) \tag{1.1a}$$

$$\mathcal{Y}_t = h_t(\mathcal{X}_t, \mathcal{W}_t, \mathbf{u}_t) \tag{1.1b}$$

dove $f_t(\cdot)$ è un sistema di equazioni alle derivate parziali e $h_t(\cdot)$ è una funzione di trasformazione d'uscita non lineare e tempo-variante.

Modelli concettuali (tempo-discreto). Il sistema $\mathcal{S}$ è descritto da un modello tempo-discreto, basato sulla concettualizzazione e semplificazione delle leggi fisiche che descrivono la dinamica del sistema:

$$\mathbf{X}_{t+1} = \mathbf{f}_t(\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t) \tag{1.2a}$$

$$\mathbf{Y}_t = \mathbf{h}_t(\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t) \tag{1.2b}$$

dove $\mathbf{f}_t(\cdot)$ è una funzione vettoriale non lineare e tempo-variante che modellizza le dinamiche di $\mathbf{X}_t$, e $\mathbf{h}_t(\cdot)$ è una funzione di trasformazione d'uscita, anch'essa non lineare e tempo-variante.

In entrambi i casi, la descrizione del sistema $\mathcal{S}$ è data da un modello PB $\mathcal{M}$ (eqs. (1.1) o (1.2)), le cui dimensionalità di stato, uscita, forzante esogena e controllo sono rispettivamente uguali a N_x, N_y, N_w and N_u. Un'accurata modellizzazione del sistema $\mathcal{S}$ si riflette spesso in una grande dimensionalità N_x, la quale, insieme a forti non linearità, può rendere il modello $\mathcal{M}$ molto esigente dal punto di vista computazionale.

1.1.3 Il problema $\mathcal{P}$

Si consideri infine che si voglia risolvere un certo problema $\mathcal{P}$ definito su $\mathcal{S}$, il quale, a seconda della sua forma, può implicare che l'interesse sia posto sulla traiettoria di $\mathbf{Y}_t$ o su di funzionale $\mathbf{J}(\cdot)$ di tale traiettoria (ad esempio, la funzione obiettivo dei problemi di pianificazione e gestione). La letteratura scientifica mostra una varietà di problemi che possono essere considerati, i cui nomi e scopi variano fortemente con il dominio scientifico di interesse. Questi problemi vengono generalmente classificati nelle seguenti categorie.

Model diagnostics Numerosi problemi possono verificarsi quando si costruisce, si valuta e si utilizza il modello $\mathcal{M}$. Questi possono essere classificati nei seguenti gruppi.

- *Identificazione..* Durante la fase di identificazione del modello $\mathcal{M}$, in particolare nel caso di modelli di grande complessità, si deve prendere in esame la relazione tra i dati e i parametri θ. Un aumento della complessità del modello si traduce infatti in un aumento del numero di parametri θ da definire e calibrare, spesso senza la raccolta di dati ulteriori (Beven, 2006). Questo può portare facilmente alla sovra-parametrizzazione e all'equifinalità (o non-unicità; von Bertalanffy, 1968; Beven, 1993), ovvero alla presenza di modelli multipli (o set di parametri) che mostrano di adattarsi allo stesso modo alle osservazioni. Un'imprecisa identificazione dei parametri può riflettersi in un'incertezza non trascurabile sull'uscita del modello $\mathbf{Y}_t$ e in un problema di identificazione mal posto (o mal condizionato; Beck, 1987), ad esempio nel caso in cui il contenuto informativo dei dati disponibili per definire il problema di identificazione non permetta l'esistenza di una soluzione matematica univoca. Per evitare tali problemi, si possono usare alcune tecniche statistiche (vedi, ad esempio, van Werkhoven *et al.*, 2009) per calcolare la discrepanza tra il contenuto informativo dei dati e il numero di parametri da calibrare.

- *Analisi di sensitività.* Il modello $\mathcal{M}$ viene utilizzato per calcolare la sensitività dell'uscita $\mathbf{Y}_t$, o del funzionale $\mathbf{J}(\cdot)$, rispetto ad una variazione di fattori selezionati, come ad esempio i parametri θ del modello, lo stato iniziale $\mathbf{X}_0$, la forzante esogena $\mathbf{W}_t$, le variabili di controllo e di pianificazione $\mathbf{u}_t$ e $\mathbf{u}^p$ (ad esempio, studiare la sensitività della produzione agricola in funzione della variazione della pedologia del suolo). Per una panoramica degli approcci e dei metodi di analisi si sensitività, vedi Saltelli *et al.* (2000).

- *Data assimilation.* Le predizioni del modello $\mathcal{M}$ circa lo stato $\mathbf{X}_t$ del sistema $\mathcal{S}$ viene combinato con le osservazioni dello stato corrente del sistema, allo scopo di produrre ciò che viene chiamato *analysis step*, che tenta di bilanciare l'incertezza contenuta nei dati e nella predizione del modello. La data assimilation, conosciuta anche con il nome di stima dello stato, viene largamente adottata nelle previsioni meteo, in idrologia e in oceanografia (vedi Kalnay, 2002; Bennett, 2002).

Pianificazione e gestione. In questi problemi di tipo decisionale, i valori delle variabili di pianificazione e di controllo $\mathbf{u}^p$ e $\mathbf{u}_t$ che ottimizzano il funzionale $\mathbf{J}(\cdot)$ vengono calcolati tramite metodi di analisi dei sistemi (vedi, ad esempio, Loucks *et al.*, 1980; Soncini-Sessa *et al.*, 2007, e bibliografia). Più precisamente, in pianificazione il valore di $\mathbf{u}^p$ che massimizza $\mathbf{J}(\cdot)$ deve essere determinato (ad esempio, progettare, all'interno di limiti appropriati, la forma di uno sfioratore), mentre nella gestione, l'insieme delle decisioni ottime $\{\mathbf{u}_1, \ldots, \mathbf{u}_H\}$ (con H pari alla lunghezza dell'orizzonte temporale considerato) che massimizza $\mathbf{J}(\cdot)$ deve essere ottenuto risolvendo un problema di controllo ottimo (ad esempio, progettare la politica di controllo di un serbatoio).

Simulazione. In questo caso, viene valutata la variazione dell'uscita $\mathbf{Y}_t$, o del funzionale $\mathbf{J}(\cdot)$, in funzione di diversi valori di $\mathbf{u}^p$ o di differenti traiettorie del controllo $\mathbf{u}_t$ o della forzante esogena $\mathbf{W}_t$ (ad esempio, studiare l'effetto del cambiamento climatico sulla dinamica di un bacino imbrifero). L'analisi di simulazione, chiamata spesso anche analisi di scenario, analisi what-if o simulazione della politica, può essere vista come un passaggio elementare e necessario all'interno di quasi tutte le categorie sopra menzionate.

Generalmente, la dimensionalità dello stato N_x (così come quelle della forzante esogena e del controllo N_w e N_u) dei modelli PB adottati per descrivere il sistema $\mathcal{S}$ è estremamente grande, e la soluzione di qualunque problema $\mathcal{P}$ è praticamente impossibile da ottenere, a causa delle sue elevate richieste computazionali. Poiché il cuore della difficoltà consiste nella dimensionalità del modello $\mathcal{M}$, la soluzione naturale è quella di identificare un modello ridotto $\tilde{m}$ in grado di simulare accuratamente l'uscita $\mathbf{Y}_t$, o il funzionale $\mathbf{J}(\cdot)$, del modello $\mathcal{M}$, ma con una dimensionalità tale da poter risolvere il problema $\mathcal{P}$. Il modello ridotto $\tilde{m}$ viene chiamato *emulation model* e sostituisce il modello $\mathcal{M}$ nel problema $\mathcal{P}$. L'idea di fondo di questa riduzione è che alcuni processi descritti dal modello PB siano più significativi di altri rispetto a $\mathbf{Y}_t$ o $\mathbf{J}(\cdot)$.

1.2 Come ridurre la complessità?

L'emulation model $\tilde{m}$ deve essere operativamente equivalente al modello $\mathcal{M}$, nel senso che deve riprodurre accuratamente l'uscita $\mathbf{Y}_t$ o il funzionale $\mathbf{J}(\cdot)$, ma deve anche essere computazionalmente efficiente, poiché deve essere impiegato per la risoluzione del problema $\mathcal{P}$. A seconda dello scopo dell'emulation model (cioè riprodurre $\mathbf{Y}_t$ oppure $\mathbf{J}(\cdot)$), le diverse tecniche disponibili in letteratura possono essere ricondotte a due approcci metodologici principali: *non-dynamic emulation modelling* e *dynamic emulation modelling*. In entrambi i casi, bisogna considerare che la validità di qualunque emulation model è condizionata alla validità del modello PB sul quale viene sviluppato. Infatti, un emulation model è consistente nel momento in cui il modello PB $\mathcal{M}$ di partenza è altamente affidabile e può essere considerato rappresentativo del sistema $\mathcal{S}$.

1.2.1 Non-dynamic emulation modelling

L'approccio del *non-dynamic emulation modelling*, introdotto per la prima volta da Blanning (1975) con il termine *meta-modello*, si basa sull'idea di identificare, su di un dataset prodotto tramite simulazione del modello $\mathcal{M}$, una funzione di regressione empirica e di basso ordine $\tilde{\mathbf{g}}(\cdot)$ che esprima in maniera esplicita la variazione del funzionale $\mathbf{J}(\cdot)$ rispetto al più piccolo subset dei parametri θ del modello PB e della variabile di pianificazione $\mathbf{u}^p$. Un meta-modello è quindi basato sull'identificazione di una funzione $\tilde{\mathbf{g}}(\cdot)$ che mappa le variabili θ e $\mathbf{u}^p$ nel funzionale $\mathbf{J}(\cdot)$. Quando $\mathbf{J}(\cdot)$ rappresenta l'obiettivo di un problema di pianificazione $\mathcal{P}$ (vedi paragrafo 1.1.3), questa metodologia è conosciuta anche come *response surface* (Box and Wilson, 1951; Kleijnen (2008)) o *surrogate-based analysis and optimization* (Queipo *et al.*, 2005).

I meta-modelli sono veloci, computazionalmente efficienti e basati su solide basi teoriche; tuttavia, non forniscono un'approssimazione dinamica dei processi descritti dal modello $\mathcal{M}$, ma piuttosto simulano l'effetto di una variazione in una variabile di pianificazione $\mathbf{u}^p$ o in un parametro θ su di un funzionale $\mathbf{J}(\cdot)$ della traiettoria dell'uscita del modello PB. Di conseguenza, i non-dynamic emulation models non possono essere utilizzati per tutti quei problemi che richiedono una descrizione dinamica, anche se approssimata, del sistema $\mathcal{S}$, come ad esempio la gestione (ovvero il controllo ottimo) o la stima dello stato.

1.2.2 Dynamic emulation modelling

Diversamente dai meta-modelli, lo scopo del dynamic emulation modelling è fornire una descrizione semplificata, ma dinamica, del modello $\mathcal{M}$. Per questa ragione, l'obiettivo del dynamic emulation modelling è garantire una riproduzione $\tilde{\mathbf{y}}_t$ dell'uscita $\mathbf{Y}_t$, utilizzando un numero ridotto di variabili di stato $\tilde{\mathbf{x}}_t$, forzanti esogene $\tilde{\mathbf{w}}_t$ e controlli $\tilde{\mathbf{u}}_t$. Il *problema di dynamic emulation modelling* d'interesse può essere definito come segue.

Dato un modello $\mathcal{M}$ nella forma dell'eq. (1.1) o dell'eq. (1.2), con dimensionalità N_x, N_w, N_u e N_y, identificare un modello ridotto $\tilde{m}$, con dimensionalità $\tilde{n}_x \ll N_x$, $\tilde{n}_w \ll N_w$, $\tilde{n}_u \ll N_u$ e $\tilde{n}_y = N_y$, della seguente forma

$$\tilde{\mathbf{x}}_{t+1} = \tilde{\mathbf{f}}_t(\tilde{\mathbf{x}}_t, \tilde{\mathbf{w}}_t, \tilde{\mathbf{u}}_t) \tag{1.3a}$$

$$\tilde{\mathbf{y}}_t = \tilde{\mathbf{h}}_t(\tilde{\mathbf{x}}_t, \tilde{\mathbf{w}}_t, \tilde{\mathbf{u}}_t) \tag{1.3b}$$

dove $\tilde{\mathbf{f}}_t(\cdot)$ è una funzione vettoriale non lineare e tempo-variante che descrive la dinamica di $\tilde{\mathbf{x}}_t$ e $\tilde{\mathbf{h}}_t(\cdot)$ è una funzione di trasformazione d'uscita non lineare e tempo-variante. Lo scopo principale del problema di emulation modelling è quindi ridurre la dimensionalità del modello PB da N_x a $\tilde{n}_x$ e da N_w a $\tilde{n}_w$. Per quanto riguarda il controllo $\mathbf{u}_t$, la sua riduzione potenziale da N_u a $\tilde{n}_u$, implica che non tutti i controlli del problema $\mathcal{P}$ sono rilevanti nell'influenzare le dinamiche dello stato e dell'uscita del sistema.

L'identificazione di un dynamic emulation model[1] $\tilde{m}$ è un processo piuttosto complesso, poiché i modelli PB, soprattutto nella modellistica ambientale, sono fortemente non lineari e caratterizzati da un'alta dimensionalità. Inoltre, manca una visione teorica condivisa, poiché approcci differenti sono stati sviluppati indipendentemente in diversi domini d'interesse. I successivi due paragrafi illustreranno una procedura generale per il *dynamic emulation modelling*, presentando i passaggi principali che sono comunemente considerati in questo problema e proponendo una tassonomia dei differenti approcci.

[1] O, semplicemente, emulation model nel seguito.

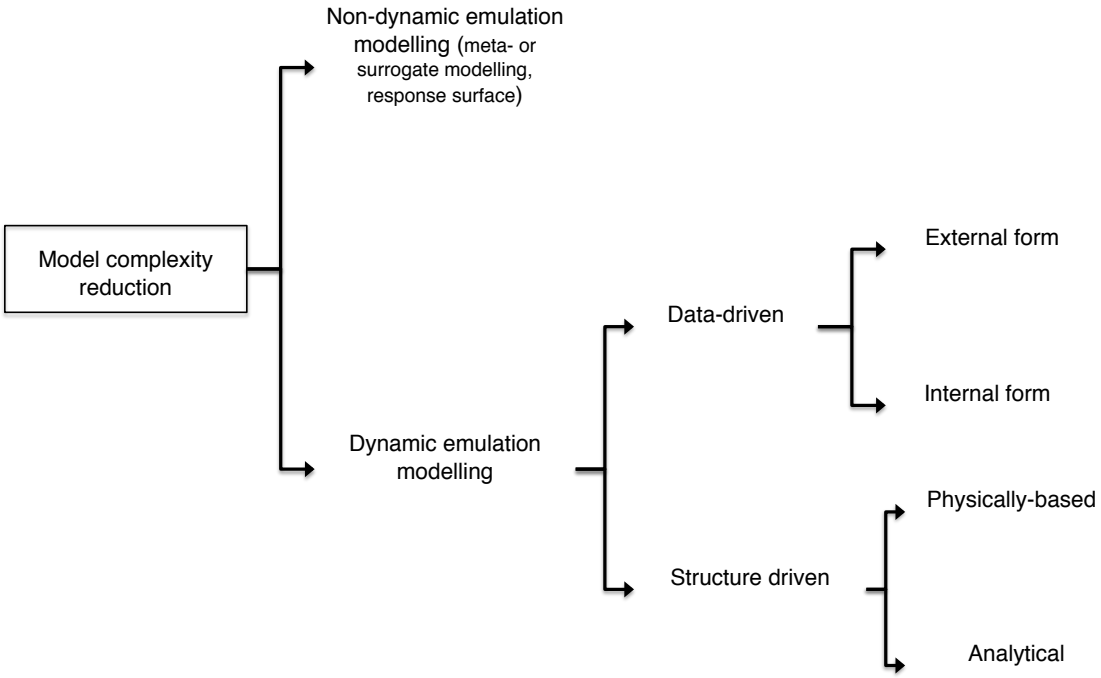

Figura 1.1: Tassonomia degli approcci per ridurre la complessità di un modello (da Galelli, 2010)

1.3 I passaggi principali del dynamic emulation modelling

Quando si considera un problema di *dynamic emulation modelling*, si devono generalmente seguire cinque passaggi principali.

1. ***Design of experiments.*** Il Design Of Experiment (DOE), o Design and Analysis of Computer Experiments (DACE), mira a definire una sequenza di esperimenti di simulazione, eseguiti tramite il modello PB, con lo scopo di ottenere il dataset da utilizzare nella fase di identificazione dell'emulation model. In particolare, il progetto degli esperimenti definisce gli ingressi al modello PB (per esempio le traiettorie della forzante esogena $\mathbf{W}_t$ e del controllo $\mathbf{u}_t$, o i valori dei parametri del modello PB e della variabile di pianificazione $\mathbf{u}^p$) che produrranno le traiettorie simulate dello stato $\mathbf{X}_t$ e dell'uscita $\mathbf{Y}_t$. L'accuratezza del DOE dipende dai differenti approcci al problema di emulation modelling (vedi prossimo paragrafo): se l'emulation model è identificato in base ai dati, il DOE deve essere accurato, in quanto dovrà fornire un dataset con un elevato contenuto informativo; se l'emulation model è derivato dalla struttura del modello PB, il DOE deve semplicemente fornire pochi esperimenti, i risultati dei quali verranno utilizzati per validare le capacità dell'emulation model. In ogni caso, il DOE definisce anche l'intervallo di campionamento utilizzato per campionare i dati generati, con un'ampia risoluzione temporale, dal modello PB. Questi intervalli di campionamento corrispondono al passo temporale dell'emulation model.

2. ***Simulation runs.*** In accordo con gli ingressi al modello PB definiti durante la fase di DOE, le simulazioni sono eseguite e i dati vengono raccolti, in base ad un intervallo di campionamento specifico, in un dataset. Se viene utilizzato un modello fisicamente basato, il dataset viene ottenuto per via numerica risolvendo un sistema di equazioni alle derivate parziali, discretizzando il dominio spazio-tempo con una certa griglia. Se viene utilizzato un modello concettuale, il modello fornisce direttamente il dataset discreto, il quale può essere post-processato come nel caso dei modelli fisicamente basati.

3. ***Riduzione* (*lumping*).** Lo scopo principale di un problema di *dynamic emulation modelling* è ridurre la dimensionalità N_x dello stato del modello PB a $\tilde{n}_x$ e, nel caso, le dimensionalità della forzante esogena e del controllo da N_w e N_u a $\tilde{n}_w$ e $\tilde{n}_u$ rispettivamente. Quando la dimensionalità è troppo elevata, o la conoscenza della struttura del modello PB non è accurata, è pratica comune ridurre la dimensionalità proiettando lo spazio di stato iniziale ad alta dimensionalità in uno stato a minore dimensionalità. Una riduzione di questo tipo si può basare su metodi analitici (tipicamente nel caso di modelli PB lineari; vedi Antoulas *et al.*, 2001), metodi numerici (come l'analisi delle componenti principali; vedi Jollife *et al.*, 1999) oppure metodi fisicamente basati (come l'aggregazione spaziale). Quest'ultima viene generalmente adottata quando il modello PB è distribuito nello spazio (cioè il dominio spaziale $\mathcal{L}$ è discretizzato in un certo numero di celle). Una volta che la riduzione è stata eseguita, la riduzione può essere ulteriormente raffinata selezionando, tra le variabili proiettate, solo quelle maggiormente rilevanti nello spiegare l'uscita $\mathbf{Y}_t$. Questo metodo si basa generalmente su tecniche di selezione delle variabili (Guyon e Elisseeff, 2003) o su considerazioni di tipo fisico.

4. ***Identificazione del modello*** Quando il passaggio precedente termina, le variabili che caratterizzano l'emulation model sono note, e le funzioni $\tilde{\mathbf{f}}_t(\cdot)$ e $\tilde{\mathbf{h}}_t(\cdot)$ che appaiono nell'eq. (1.3) devono essere definite. Se il modello viene costruito a partire da dati prodotti tramite esperimenti di simulazione, questo passaggio non è altro che un problema tradizionale di identificazione del modello, e richiede la selezione della classe di modello per $\tilde{\mathbf{f}}_t(\cdot)$ e $\tilde{\mathbf{h}}_t(\cdot)$, oltre alle fasi di calibrazione e validazione. In alternativa, se l'emulation model viene derivato dalla struttura del modello PB, la definizione delle due funzioni è il risultato 'automatico' della semplificazione del modello PB. In ogni caso, l'emulation model è validato sfruttando i dati prodotti tramite il modello PB.

5. ***Interpretazione fisica*** A seconda del modo in cui l'emulation model è stato costruito e del suo successivo utilizzo nel problema $\mathcal{P}$, può essere data un'interpretazione di tipo fisico (cioé l'emulation model fornisce una descrizione che ha una rilevanza diretta sulla realtà fisica del sistema $\mathcal{S}$ in esame). Questo si fa tipicamente analizzando la struttura dell'emulation model o i suoi parametri,

che dovrebbero avere una qualche rilevanza fisica.

Questi passaggi non vengono considerati solo una volta durante l'identificazione dell'emulation model, in quanto possono sorgere un certo numero di ricorsioni. Se le prestazioni del modello, ad esempio, non sono soddisfacenti, si possono progettare nuovi esperimenti di simulazione, oppure valutare diversi approcci di riduzione.

1.4 Tassonomia

Un gran numero di tecniche differenti è disponibile per problemi di *dynamic emulation modelling*. La distinzione principale è tra approcci *data-based* e *structure-based*; entrambe le tipologie di approcci sfruttano in modo diverso i cinque passaggi descritti nel paragrafo precedente.

Approccio data-based. Include tutte le tecniche che si basano sull'utilizzo di un dataset discreto $\mathcal{F}$ di tuple $\{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{X}_{t+1}\}$ (con $t \in [1, H]$) ottenuto tramite simulazione del modello PB su di un dato orizzonte H. Le diverse tecniche di riduzione appartenenti a questo approccio possono essere classificate in due gruppi differenti, in funzione della forma (esterna o interna) dell'emulation model che costruiscono.

- *Forma esterna.* Le techniche appartenenti a questo gruppo propongono di simulare l'uscita del modello $\mathbf{Y}_t$ con un emulation model in forma esterna, semplificando quindi il problema di dynamic emulation modelling dell'eq. (1.3) con la seguente espressione

$$\tilde{\mathbf{y}}_t = \tilde{\mathbf{f}}_t(\tilde{\mathbf{y}}_{t-1}, \ldots, \tilde{\mathbf{y}}_{t-q}, \tilde{\mathbf{w}}_t, \tilde{\mathbf{u}}_t) \tag{1.4}$$

  dove $\tilde{\mathbf{f}}_t(\cdot)$ è una funzione vettoriale non lineare e tempo-variante che modellizza $\tilde{\mathbf{y}}_t$, e p denota il numero di termini auto-regressivi. La struttura del modello, così come la classe del modello per $\tilde{\mathbf{f}}_t(\cdot)$, sono identificati in base al dataset $\mathcal{F}$ prodotto dal modello PB. Il vantaggio principale di questo approccio è il fatto che la dimensionalità dello stato $\tilde{n}_x$ diviene pari a zero, e l'uscita del modello $\tilde{\mathbf{y}}_t$ è spiegata come una funzione di termini auto-regressivi, forzanti esogene e variabili di controllo. Diviene

perciò utile nel momento in cui non è necessaria una descrizione, anche se semplificata, della dinamica dello stato del modello PB[2]. Di recente, sono apparse alcune applicazioni in campo ambientale; vedi, ad esempio, Young e Ratto (2010), Galelli *et al.* (2009), Galelli *et al.* (2008), Ratto *et al.* (2007), Young (1999).

- *Forma interna.* Le tecniche di questo secondo gruppo seguono il problema di dynamic emulation modelling dell'eq. (1.3), e mirano ad identificare un emulation model $\tilde{m}$ in forma interna. In altre parole, simulano il comportamento dell'uscita $\mathbf{Y}_t$ utilizzando il minor numero di variabili caratterizzanti il modello originale $\mathcal{M}$, selezionando il più piccolo sottoinsieme di variabili all'interno di $\mathbf{X}_t$, $\mathbf{W}_t$ e $\mathbf{u}_t$. Anche in questo caso, la selezione di questi sottoinsiemi e l'identificazione delle funzioni $\tilde{\mathbf{f}}_t(\cdot)$ e $\tilde{\mathbf{h}}_t(\cdot)$ si basano sul dataset $\mathcal{F}$. Per applicazioni di queste tecniche a problemi di modellizzazione ambientale, vedi, ad esempio, Castelletti *et al.* (2010c), Galelli *et al.* (2010), Yang *et al.* (2007).

Approccio structure-based. Include tutte le tecniche che derivano un emulation model 'manipolando' la struttura del modello PB. Generalmente, questo approccio viene impiegato ogni volta che l'uscita $\mathbf{Y}_t$ non è definita oppure, in maniera equivalente, coincide con il vettore di stato $\mathbf{X}_t$. In questo caso, il criterio adottato per costruire l'emulation model è quello di descrivere i processi nel modello $\mathcal{M}$ utilizzando un numero ridotto di variabili di stato. In altre parole, ciò significa sostituire il vettore di stato $\mathbf{X}_t$ (con dimensionalità N_x) con un vettore di stato ridotto $\tilde{\mathbf{x}}_t$ (con dimensionalità $\tilde{n}_x \ll N_x$) in modo che la varianza dei dati originali venga per quanto possibile preservata. Le tecniche appartenenti a questo approccio possono essere classificate in due gruppi differenti.

- *Fisicamente basato.* Queste tecniche si basano su un'analisi dettagliata della struttura del modello PB e sulla sua successiva semplificazione euristica, rimuovendo variabili o meccanismi di feedback (vedi, per esempio, Van Nes e Scheffer, 2005; Cox *et al.*, 2006; Crout *et al.*, 2009). L'utilizzo di

[2]Altrimenti, bisogna considerare un problema di realizzazione minima.

queste tecniche è limitato ad esperti, poiché richiede una certa interazione da parte dell'utente, ed è comunemente associato a scopi puramente scientifici, in quanto preserva l'interpretabilità fisica del modello PB, aumentando quindi la conoscenza del modo in cui i modelli complessi generano i loro risultati.

- *Analitico.* Queste tecniche propongono una riduzione del vettore di stato da $\mathbf{X}_t$ a $\tilde{\mathbf{x}}_t$ proiettando il sistema di equazioni di transizione di stato originale in un sotto-spazio di minori dimensioni. Se il modello PB è lineare (o debolmente non lineare), sono disponibili differenti metodi di proiezione esatta, basati sulla *Singular Value Decomposition* o sul *Moment Matching* (vedi Antoulas *et al.*, 2001). Questi metodi vengono comunemente adottati per lo studio di processi industriali (vedi, Okino e Mavrovouniotis, 1998; Shvartsman e Kevrekidis, 1998). Al contrario, se il modello PB è non lineare, alcuni autori (vedi, per esempio, Bernhardt, 2008; van der Merwe *et al.*, 2007) propongono per prima cosa di proiettare il sistema di equazioni di transizione di stato originale in un sotto-spazio di minore dimensione con dimensionalità $\tilde{n}_x$, per poi identificare una relazione ingresso-uscita che descriva la dinamica di $\tilde{\mathbf{x}}_t$ come una funzione della forzante esogena $\tilde{\mathbf{w}}_t$, del controllo $\tilde{\mathbf{u}}_t$ e dei valori di $\tilde{\mathbf{x}}$ negli istanti di tempo precedenti (ad esempio $\tilde{\mathbf{x}}_{\mathbf{t-1}}$, $\tilde{\mathbf{x}}_{\mathbf{t-2}}$, ecc.). Con quest'ultimo approccio, potrebbero non essere garantite le caratteristiche di un processo markoviano per gli elementi presenti in $\tilde{\mathbf{x}}_t$.

Capitolo 2

Un approccio procedurale al dynamic emulation modelling

Si consideri il *problema di dynamic emulation modelling* descritto nel paragrafo 1.2.2. Dato un modello $\mathcal{M}$ (fisicamente basato, eq. (1.1), o concettuale, eq. (1.2)) con dimensionalità dello stato, dell'uscita, della forzante esogena e del controllo rispettivamente pari a N_x, N_y, N_w eN_u, puntiamo ad identificare un dynamic emulation model $\tilde{m}$ in forma interna (eq. (1.3)), con dimensionalità dello stato, dell'uscita, della forzante esogena e del controllo rispettivamente pari a $\tilde{n}_x \ll N_x$, $\tilde{n}_y = N_y$, $\tilde{n}_w \ll N_w$ e $\tilde{n}_u \ll N_u$, il cui output $\tilde{\mathbf{y}}_t$ riproduca accuratamente l'output $\mathbf{Y}_t$. Un tale problema di identificazione è guidato dalla necessità di ridurre i requisiti di calcolo del modello PB, i quali impediscono la risoluzione pratica del problema $\mathcal{P}$ (vedi paragrafo 1.2).
Questo paragrafo propone un approccio procedurale (Galelli, 2010), composto da sette fasi, per l'identificazione di *data-based emulation model* (vedi Figura 2.1).

2.1 Concettualizzazione del problema

Il punto più importante di questa fase è l'analisi del problema $\mathcal{P}$, che deve essere risolto tramite l'adozione del modello $\tilde{m}$. In particolare, questa analisi si concentra sull'uscita $\mathbf{Y}_t$ che deve essere approssimata dall'uscita dell'emulation model $\tilde{\mathbf{y}}_t$. La selezione di questa variabile non è una prerogativa di questa fase, in quanto è in qualche modo fornita dal problema $\mathcal{P}$ che deve essere risolto. Si consideri, ad esempio, il seguente problema di gestione: progettare la politica di controllo per una serie di mixer, impiegati per minimizzare, aumentando l'ossigenazione in profondità, la

concentrazione media di Manganese disciolto nello strato di fondo di un serbatoio. La soluzione di questo problema di controllo ottimo si basa sulla definizione dell'obiettivo di gestione e del corrispondente indicatore per passo[1], che può quindi essere considerato come l'uscita naturale $\mathbf{Y}_t$ del modello PB. Quando il sistema $\mathcal{S}$ è affetto non solo dalla forzante esogena $\mathbf{W}_t$, ma anche da una variabile di controllo $\mathbf{u}_t$ (come la potenza dei mixer nell'esempio precedente), l'analisi del problema $\mathcal{P}$ fornisce anche il valore del passo decisionale Δt. L'informazione circa questo valore verrà impiegata successivamente nelle fasi seguenti (fase 2 - *Desing Of Experiments* e fase 3 - *Simulation Runs*).

Durante questa fase, il modello PB è analizzato, con lo scopo di conoscere la fisica descritta dal sistema. Eventualmente, tale conoscenza può essere espressa tramite una rete causale di tutti i processi modellizzati, o solamente dei processi che portano all'uscita $\mathbf{Y}_t$ (in quanto alcuni processi descritti dal modello PB non sono significativi nello spiegare $\mathbf{Y}_t$).

2.2 Design Of Experiments

Lo scopo del Design Of Experiments (DOE) è progettare una sequenza di esperimenti di simulazione, espressi in termini di *factors* settati a valori predefiniti, che verranno successivamente eseguiti tramite il modello PB per ottenere un dataset per l'identificazione dell'emulation model.

Quando si considera un problema di *non-dynamic emulation modelling*, questi fattori sono i parametri θ del modello PB e le variabili di pianificazione $\mathbf{u}^p$, i cui valori rimangono costanti per l'intera simulazione. Per questo motivo, un DOE può essere facilmente rappresentato tramite una matrice, dove le righe sono gli esperimenti (simulazioni), e le colonne denotano i settaggi dei *factors*. Poiché il numero degli esperimenti può aumentare rapidamente con il numero di *factors* e dei livelli[2], varie ricerche hanno sviluppato diverse tecniche statistiche (ad esempio, orthogonal ar-

[1] Un indicatore per passo associato al tempo t esprime il costo prodotto dalla transizione di stato da $\mathbf{X}_t$ a $\mathbf{X}_{t+1}$ e dipende da tutte le variabili relative all'intervallo temporale $[t, t+1)$ (vedi Soncini-Sessa *et al.*, 2007).

[2] In un *full factorial design*, ad esempio, sono considerate tutte le combinazioni dei *factors* e dei livelli corrispondenti: il numero di esperimenti dettati da questo tipo di progetto è il prodotto del numero dei livelli per ogni fattore. L'ampiezza di un esperimento fattoriale completo aumenta esponenzialmente con il numero dei fattori, il che porta facilmente ad un numero ingestibile di esperimenti (Simpson *et al.*, 2001).

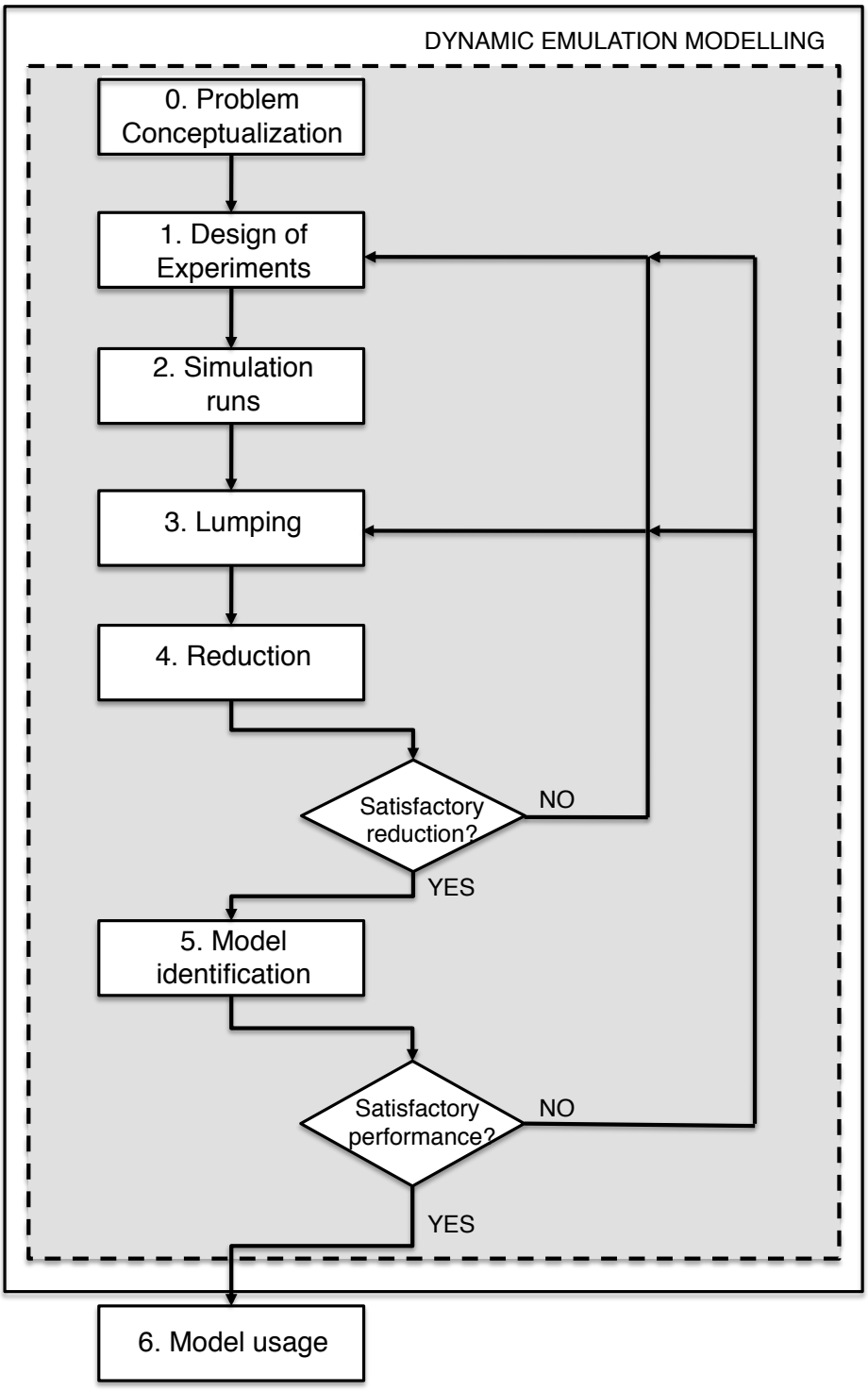

Figura 2.1: Le fasi della procedura di dynamic emulation modelling (da Galelli, 2010).

rays, latin hypercube, space filling design) allo scopo di campionare efficacemente lo spazio dei *factors* tenendo conto del numero limitato di simulazioni che si possono eseguire (per una panoramica di queste tecniche, vedi, Kleijnen, 2008; Kleijnen *et al.*, 2005; Montgomery, 2000).

Al contrario, quando si considera un problema di *dynamic emulation modelling*, i *factors* del DOE sono la forzante esogena $\mathbf{W}_t$ e la variabile di controllo $\mathbf{u}_t$, i cui valori cambiano periodicamente durante l'intera sessione di simulazione. In questo caso, il DOE è il piano di campionamento nello spazio delle forzanti $\mathbf{W}_t$ e dei controlli $\mathbf{u}_t$, e il suo obiettivo è conoscere il più possibile circa il comportamento del modello PB, esplorando, tramite simulazione, l'area più ampia possibile all'interno degli spazi $\mathcal{L}_{\mathbf{Y}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$ e $\mathcal{L}_{\mathbf{X}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$. Poiché una singola sessione di simulazione richiede la definizione di una serie temporale di valori (traiettoria) sia per $\mathbf{W}_t$ che per $\mathbf{u}_t$ sull'intero orizzonte di simulazione H, determinare tali traiettorie significa campionare dei valori numerici per ogni valore delle forzanti e dei controlli negli spazi $\mathcal{L}_{\mathbf{W}_t}$ e $\mathcal{L}_{\mathbf{u}_t}$. Per quanto riguarda l'intervallo di campionamento di $\mathbf{u}_t$, questo è dato automaticamente dal passo decisionale Δt definito nella fase precedente; mentre per quanto riguarda l'intervallo di campionamento di $\mathbf{W}_t$, questo dipende dalla disponibilità dei dati e dai requisiti del modello PB. Considerando le severe limitazioni sul numero di simulazioni a causa dei requisiti di calcolo del modello PB, devono essere impiegate tecniche appropriate per selezionare le traiettorie che consentano di esplorare in maniera efficace gli spazi $\mathcal{L}_{\mathbf{Y}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$ e $\mathcal{L}_{\mathbf{X}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$. Diversamente dal *non-dynamic emulation modelling*, non esiste una procedura generale consolidata per il DOE, ed è pratica comune basare gli esperimenti di simulazione su considerazioni fisiche e conoscenza a priori (vedi, ad esempio, Galelli *et al.*, 2010; Castelletti *et al.*, 2010c).

Siccome il modello PB verrà eseguito con un passo molto breve per garantire un certo grado di accuratezza numerica (vedi fase 3 - *Simulation Runs*), durante il DOE viene definito un intervallo di campionamento (*aggregazione temporale*), in modo da ridurre il numero di campioni che andranno a comporre il dataset discreto $\mathcal{F}$. Questa aggregazione temporale, progettata per diminuire la ridondanza in $\mathcal{F}$, deve essere interamente appropriata per l'identificazione dell'emulation model. La selezione dell'intervallo di campionamento è generalmente dettata da conoscenze a priori o dal problema $\mathcal{P}$ (ad esempio, può corrispondere al passo decisionale Δt). È interessante notare che questa aggregazione temporale definisce automaticamente il passo di modellizzazione dell'emulation model.

2.3 Simulation Runs

Una volta concluso il DOE, la sequenza degli esperimenti di simulazione deve essere eseguita tramite il modello PB. Se si utilizza un modello fisicamente basato, viene usato un esplicito schema di integrazione numerica delle equazioni alle derivate parziali, e la griglia spazio-tempo deve rispettare un vincolo di risoluzione minima, in modo da evitare problemi di *numerical diffusion*, e di conseguenza la divergenza dello schema di integrazione numerico. La condizione di Courant - Friedrichs - Lewy (condizione CFL; Courant *et al.*, 1967), ad esempio, è una condizione necessaria, ma in generale non sufficiente, per la convergenza nella risoluzione di equazioni alle derivate parziali iperboliche, che vengono comunemente impiegate nella meccanica dei fluidi[3]. Se viene utilizzato un modello concettuale, il passo di simulazione è definito semplicemente dalle caratteristiche del modello. In entrambi i casi, i dati che compongono il dataset discreto $\mathcal{F}$ delle tuple $\{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{X}_{t+1}\}$ (con $t \in [1, H]$) sono raccolti tramite l'intervallo di campionamento selezionato durante il DOE, indipendentemente dalla risoluzione della griglia spazio-tempo del modello PB o dal passo di simulazione.

2.4 Lumping

Siccome il cuore di un problema di *dynamic emulation modelling* è di ridurre la dimensionalità del modello PB da N_x, N_w e N_u a $\tilde{n}_x$, $\tilde{n}_w$ e $\tilde{n}_u$, la fase cruciale è rappresentata dalla selezione, dai vettori $\mathbf{X}_t$, $\mathbf{W}_t$ e $\mathbf{u}_t$, dei sottoinsiemi $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ delle variabili che andranno a costituire gli argomenti dell'eq. (1.3). Nelle applicazioni ai casi reali il valore di N_x è comunemente molto grande, poiché l'uso pratico di modelli PB 2D-3D spesso richiede di discretizzare il dominio spaziale $\mathcal{L}$ con un numero di celle dell'ordine di $10^4 \sim 10^5$. Per quanto riguarda N_w e N_u, i loro valori sono piccoli, se confrontati con N_x, ma possono essere comunque dell'ordine di 10^2 (in particolare N_w, poiché $\mathbf{W}_t$ è spesso distribuito nello spazio). Ciò significa che poche variabili (circa dell'ordine di 10^1) appartenenti a $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ devono essere selezionate all'interno di $10^4 \sim 10^5$ variabili candidate. Questa operazione è concettualmente ben definita e, in linea di principio, progettare o impiegare degli algoritmi

[3]Per esempio, se un'onda sta attraversando una griglia discreta, il passo temporale deve essere minore del tempo impiegato dall'onda per attraversare due punti di griglia adiacenti. Come corollario, quando la separazione dei punti di griglia viene ridotta, il limite superiore per il passo temporale decresce a sua volta.

a questo proposito non è affatto proibitivo. Tuttavia, esistono alcune difficoltà nell'adottare questo approccio: in primo luogo, il numero delle variabili candidate è talmente grande che le richieste di calcolo di qualsiasi algoritmo risultano proibitive; in secondo luogo, dare un'interpretazione fisica ai risultati si rivela essere un'operazione pressoché impossibile.

Per questo motivo, lo scopo della fase di *lumping* è di trasformare i vettori $\mathbf{X}_t$ e $\mathbf{W}_t$ di variabili distribuite nello spazio, in vettori di dimensioni ridotte $\mathbf{x}_t$ e $\mathbf{w}_t$ (con dimensionalità $n_x \ll N_x$ e $n_w \ll N_w$), all'interno dei quali selezionare gli elementi appartenenti a $\tilde{\mathbf{x}}_t$ e $\tilde{\mathbf{w}}_t$. Per quanto riguarda $\mathbf{u}_t$, non è richiesta una trasformazione, poiché si assume che il controllo non sia distribuito nello spazio (e perciò $n_u = N_u$). Allo scopo di preservare l'interpretabilità fisica dei vettori $\mathbf{x}_t$ e $\mathbf{w}_t$, così come per mantenere le caratteristiche di un processo markoviano per gli elementi in $\mathbf{x}_t$, non possono essere utilizzati i metodi classici di proiezione, come l'analisi delle componenti principali (Jollife *et al.*, 1999; per un'applicazione ad un caso reale, vedi van der Merwe *et al.*, 2007), e la fase di lumping si affida quindi all'*aggregazione spaziale* degli elementi che appartengono a $\mathbf{X}_t$ e $\mathbf{W}_t$. I vettori $\mathbf{x}_t$ e $\mathbf{w}_t$ sono perciò ottenuti tramite le seguenti espressioni

$$\mathbf{x}_t = \Psi_{\Omega_t}\left[\mathbf{X}_t\right] \tag{2.1a}$$

$$\mathbf{w}_t = \Psi_{\Omega_t}\left[\mathbf{W}_t\right] \tag{2.1b}$$

dove Ψ 'e un operatore integrale che agisce su una regione appropriata Ω_t (eventualmente tempo-variante) del dominio spaziale del modello PB. Un'utile suggerimento circa la definizione di Ψ e Ω_t può provenire da una conoscenza a priori del sistema modellizzato e delle caratteristiche del problema $\mathcal{P}$. Si consideri lo stesso esempio pratico del paragrafo 2.1: siccome l'uscita del modello PB è la concentrazione media di Manganese disciolto nello strato di fondo del serbatoio, un'aggregazione spaziale 'ragionevole' è basata su Ψ e Ω_t che corrispondono rispettivamente all'operatore media e a quella regione del modello PB che corrisponde alle celle appartenenti allo strato di fondo. In alternativa, quando non si può sfruttare la conoscenza a priori, si possono usare tecniche automatiche di aggregazione spaziale, come il clustering (vedi Jain *et al.*, 1999). In ogni caso, bisogna considerare che la definizione della regione Ω_t è condizionata all'aggregazione temporale definita durante il DOE. L'ampiezza di Ω_t deve infatti essere tale che un qualsiasi segnale che attraversa tale regione impieghi una quantità di tempo confrontabile con il passo temporale dell'emulation model (cioè l'intervallo di campionamento del DOE).

Quando il lumping è terminato, i vettori disponibili per la prossima fase (*riduzione*) sono $\mathbf{x}_t$, $\mathbf{w}_t$ e $\mathbf{u}_t$ (con dimensionalità $n_x \ll N_x$, $n_w \ll N_w$ e $n_u = N_u$), mentre il dataset $\mathcal{F}$ è ora composto dalle tuple $\{\mathbf{x}_t, \mathbf{w}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{x}_{t+1}\}$ (con $t \in [1, H]$).

2.5 Riduzione

Basandosi sull'informazione contenuta nel dataset $\mathcal{F}$, lo scopo della fase di *riduzione* è selezionare, dai vettori $\mathbf{x}_t$, $\mathbf{w}_t$ e $\mathbf{u}_t$, i sottoinsiemi $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ che costituiranno gli argomenti della funzione di trasformazione d'uscita e dell'equazione di transizione di stato dell'emulation model (vedi eq. (1.3)). La riduzione deve essere tale che l'uscita $\mathbf{Y}_t$ venga riprodotta accuratamente, e, al contempo, la dimensionalità dell'emulation model $\tilde{n}_x$, $\tilde{n}_w$ e $\tilde{n}_u$ sia minima. L'approccio più semplice a questo problema è basato su considerazioni fisiche, che si affidano sulla conoscenza a priori sul sistema $\mathcal{S}$ e sul problema $\mathcal{P}$. Tuttavia, questo approccio è raramente percorribile, in quanto le dipendenze non lineari e le ridondanze tra le variabili, così come la loro numerosità, (dopo il lumping la dimensionalità è dell'ordine di circa 10^2), rendono il problema di riduzione ingestibile da un operatore umano. Per superare queste difficoltà, si scopre essere utile l'adozione di algoritmi di selezione delle variabili (o *features*), che, dato un dataset $\mathcal{F}$ e una variabile di output $\mathbf{Y}_t$ da spiegare, selezionano automaticamente le variabili più rilevanti all'interno di un set di variabili candidate (Guyon e Elisseeff, 2003). Il prossimo capitolo sarà interamente dedicato a questo approccio, proponendo due algoritmi di features selection progettati per scopi di emulation modelling.
Una volta che la fase di riduzione è terminata, i regressori (o ingressi) $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ e le uscite $\tilde{\mathbf{x}}_{t+1}$ e $\tilde{\mathbf{y}}_t$ dell'eq. (1.3), così come le dimensionalità dell'emulation model $\tilde{n}_x$, $\tilde{n}_w$ e $\tilde{n}_u$, sono noti. Se il risultato non è soddisfacente, (per esempio,la dimensionalità è ancora troppo elevata o alcune variabili ritenute rilevanti non sono state selezionate), è necessario riconsiderare l'aggregazione spaziale o temporale, ritornando quindi alla fase 4 (*Lumping*) o alla fase 2 (*Design Of Experiments*).

2.6 Identificazione dell'emulation model

In questa fase, vengono costruite le funzioni $\tilde{\mathbf{h}}_t(\cdot)$ e $\tilde{\mathbf{f}}_t(\cdot)$. Si tratta di un tradizionale problema di identificazione del modello, composto dalle fasi di selezione della classe del modello, calibrazione e validazione (o cross-validazione), che vengono effettuate basandosi sul dataset $\mathcal{F}$ (Box e Jenkins, 1970). Alcuni suggerimenti circa la selezione

Tabella 2.1: Rappresentazione schematica delle variabili e delle dimensionalità del modello PB e dell'emulation model durante le fasi 2, 3 e 4 della procedura di *dynamic emulation modelling*

fase	stato		forzante es.		controllo		uscita	
	var.	dim.	var.	dim.	var.	dim.	var.	dim.
2. simul.	$\mathbf{X}_t$	N_x	$\mathbf{W}_t$	N_w	$\mathbf{u}_t$	N_u	$\mathbf{Y}_t$	N_y
3. lumping	$\mathbf{x}_t$	$n_x \ll N_x$	$\mathbf{w}_t$	$n_w \ll N_w$	$\mathbf{u}_t$	$n_u = N_u$	$\mathbf{Y}_t$	$n_y = N_y$
4. riduzione	$\tilde{\mathbf{x}}_t$	$\tilde{n}_x \ll n_x$	$\tilde{\mathbf{w}}_t$	$\tilde{n}_w \ll n_w$	$\tilde{\mathbf{u}}_t$	$\tilde{n}_u \ll n_u$	$\tilde{\mathbf{y}}_t$	$\tilde{n}_y = n_y$

della classe del modello giungono dalla fase di riduzione, che descrive la tipologia di relazione tra i regressori $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ e gli output $\tilde{\mathbf{x}}_{t+1}$ e $\tilde{\mathbf{y}}_t$. Se, per esempio, è emersa una relazione lineare tra queste variabili, la scelta della classe del modello cadrà su di un regressore lineare. Al contrario, se la relazione è non lineare, è disponibile una grande varietà di classi differenti, ad esempio le reti neurali artificiali (vedi Bishop, 2004), modelli meccanicistici basati sui dati (Young *et al.*, 2002; Young, 1998), metodi basati sugli alberi (Breiman *et al.*, 1984), ecc.

Per quanto riguarda la calibrazione del modello, che viene eseguita in funzione della classe del modello selezionata, va specificato che deve essere consistente con lo scopo finale dell'emulation model (cioè la soluzione del problema $\mathcal{P}$). Se, per esempio, l'emulation model verrà utilizzato per la previsione ad un passo, l'indice di prestazione ottimizzato dall'algoritmo di calibrazione deve essere una certa funzione dei dati previsti dall'emulation model e simulati dal modello PB. D'altra parte, se l'emulation model verrà utilizzato in simulazione, l'indice di prestazione dovrà tener conto della differenza tra i dati simulati dall'emulation model e i dati simulati dal modello PB.

Una volta calibrato l'emulation model, le sue prestazioni in validazione sono confrontate con i dati prodotto dal modello PB. Se le prestazioni misurate[4] (ad esempio, il coefficiente di terminazione R^2) non sono soddisfacenti, si possono investigare tre diverse cause: *i*) la classe del modello selezionata non rispecchia adeguatamente la relazione tra i regressori e gli output. In questo caso, la selezione di una diversa classe può risolvere il problema. *ii*) il contenuto informativo del $\mathcal{F}$ è troppo povero ed è bisogna quindi arricchirlo riformulando il design of experiments (fase 2). *iii*)

[4]Allo scopo di valutare la validità dell'emulation model, possono essere considerati differenti criteri, oltre alla sua capacità di simulare le dinamiche del modello PB, come ad esempio la sua parsimoniosità (ovvero la capacità di ridurre la complessità del modello PB).

i regressori non sono realmente rilevanti nello spiegare l'uscita del modello PB a causa di un'errata aggregazione spaziale o temporale. Queste ultime dovrebbero essere riconsiderate, ritornando alla fase 4 (*Lumping*) o alla fase 2 (*Design Of Experiments*). Quando le prestazioni misurate sono ritenute soddisfacenti, può essere data un'interpretazione fisica e l'emulation model è pronto per il suo utilizzo finale.

2.7 Utilizzo dell'emulation model

In quest'ultima fase, l'emulation model è utilizzato per la risoluzione del problema $\mathcal{P}$. Eventualmente, la soluzione così ottenuta può essere simulata tramite il modello PB, aumentando così l'affidabilità della soluzione stessa. Si consideri lo stesso esempio pratico del paragrafo 2.1 e 2.4: l'emulation model, identificato in base ai dati generati dal modello PB, può essere usato per progettare la politica di controllo dei mixer (risolvendo un problema di controllo ottimo), il cui effetto sul sistema $\mathcal{S}$ è infine valutato tramite simulazione con il modello PB.

2.8 L'algoritmo di Recursive Feature Selection

Una volta che è stato eseguito il *lumping* delle variabili del modello PB (vedi par. 2.4), il sistema è descritto dai vettori $\mathbf{x}_t$, $\mathbf{w}_t$, $\mathbf{u}_t$ e $\mathbf{Y}_t$ (con dimensionalità n_x, n_w, n_u e $n_y = N_y$), dai quali deve essere derivato l'emulation model $\tilde{m}$. Questo compito richiede di identificare le features $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$, $\tilde{\mathbf{u}}_t$, che la dinamica dell'output $\mathbf{Y}_t$ sia accuratamente riprodotta e le loro dimensioni $\tilde{n}_x$, $\tilde{n}_w$, $\tilde{n}_u$ siano minime. Queste features verranno identificate scegliendo gli insiemi[5] $\tilde{\mathbb{X}}$, $\tilde{\mathbb{W}}$, $\tilde{\mathbb{U}}$ delle loro componenti in $\mathbb{X}$, $\mathbb{W}$, $\mathbb{U}$.

Poiché si cerca un emulation model in forma interna (vedi eq. (1.3)), bisogna identificare: *i*) la funzione di trasformazione d'uscita $\tilde{\mathbf{h}}_t(\cdot)$ e le features $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$, $\tilde{\mathbf{u}}_t$ che costituiscono i suoi argomenti; *ii*) la funzione di transizione di stato $\tilde{\mathbf{f}}_t(\cdot)$ che spiega la dinamica delle features $\tilde{\mathbf{x}}_t$ in funzione delle forzanti $\tilde{\mathbf{w}}_t$ e dei controlli $\tilde{\mathbf{u}}_t$. Come spiegato nel capitolo 2 (par. 2.5 e 2.6), questa identificazione è composta da due fasi distinte: in primo luogo, nella fase 4 (*Riduzione*), le features sono selezionate tramite l'algoritmo di *Recursive Feature Selection* (RFS, Galelli, 2010), proposto

[5]Dato un vettore $\mathbf{v}_t$, denoteremo con il simbolo $\mathbb{V}$ l'insieme delle sue componenti; di conseguenza, la cardinalità di $\mathbb{V}$ rappresenta la dimensionalità di $\mathbf{v}_t$.

nella Tabella 2.2, che fornisce gli insiemi $\tilde{\mathbb{X}}$, $\tilde{\mathbb{W}}$, $\tilde{\mathbb{U}}$; nella fase 5 (*Identificazione*), sono identificate le funzioni $\tilde{\mathbf{h}}_t(\cdot)$ e $\tilde{\mathbf{f}}_t(\cdot)$.
Le espressioni chiave nell'algoritmo RFS nella Tabella 2.2 sono le seguenti:

$$y_t^i = \tilde{h}_t^i(\tilde{\mathbf{x}}_t^{k,i}, \tilde{\mathbf{w}}_t^{k,i}, \tilde{\mathbf{u}}_t^{k,i}) \tag{2.2a}$$

nel Passo 0, e

$$\tilde{x}_{t+1}^{k,j} = \tilde{f}_t^j(\tilde{\mathbf{x}}_t^{k,j}, \tilde{\mathbf{w}}_t^{k,j}, \tilde{\mathbf{u}}_t^{k,j}) \tag{2.2b}$$

nei Passi 1 e 3. L'operazione elementare dell'algoritmo è l'identificazione dell'insieme delle features (ovvero i regressori che costituiscono gli argomenti delle funzioni nella parte destra dell'espressione precedente) che sono rilevanti per calcolare un dato output (cioè una componente di $\mathbf{Y}_t$ o una componente di $\mathbf{x}_t$). Le features devono essere selezionate tra le componenti di $\mathbf{x}_t$, $\mathbf{w}_t$, $\mathbf{u}_t$. Di conseguenza, l'implementazione dell'algoritmo RFS richiede uno strumento per selezionare le features rilevanti.

2.9 Feature selection

La selezione delle features più rilevanti, o regressori, per spiegare un certo output, richiede di formulare e risolvere un problema di *feature selection*. Questo paragrafo fornisce una breve panoramica dei diversi metodi di feature selection e propone un nuovo algoritmo (Galelli, 2010), denominato *Iterative Feature Ranking* (IFR).

2.9.1 Panoramica dei metodi di feature selection

Uno dei passi più importanti nell'identificazione di un modello data-based è la selezione delle features significative da utilizzare come regressori, in quanto non tutte le features candidate sono in generale caratterizzate dalla stessa rilevanza rispetto alla variabile di output da modellizzare. Inoltre, alcune delle features candidate possono essere correlate tra loro, generando perciò un problema di ridondanza. L'inclusione di features ridondanti e irrilevanti è causa di complessità del modello che può fortemente influenzare l'accuratezza e l'affidabilità del modello stesso. In più, all'aumentare della dimensionalità del vettore di features selezionate, il rischio di over-fitting diviene più probabile e la calibrazione del modello si rivela più difficile. La feature selection è dunque un compito essenziale, ma anche piuttosto complesso. La letteratura presenta un'ampia varietà di metodi di features selection, i quali possono essere ricondotti agli approcci *model-free* e *model-based* (Maier *et al.*, 2010).

Tabella 2.2: Algoritmo di Recursive Feature Selection.

Passo 0. Porre $k = 0$.

- Per $i = 1, \ldots, N_y$, identificare i sottoinsiemi $\tilde{\mathbb{X}}^{k,i}$, $\tilde{\mathbb{W}}^{k,i}$, $\tilde{\mathbb{U}}^{k,i}$ degli insiemi $\mathbb{X}$, $\mathbb{W}$, $\mathbb{U}$ che sono rilevanti per calcolare l'i-esima componente y_t^i di $\mathbf{y}_t$, cioè tale che esista una funzione $\tilde{h}_t^i(\cdot)$ tale che $y_t^i = \tilde{h}_t^i(\tilde{\mathbf{x}}_t^{k,i}, \tilde{\mathbf{w}}_t^{k,i}, \tilde{\mathbf{u}}_t^{k,i})$.
- Sia $\tilde{\mathbb{X}}^k = \bigcup_i \tilde{\mathbb{X}}^{k,i}$, $\tilde{\mathbb{W}}^k = \bigcup_i \tilde{\mathbb{W}}^{k,i}$, $\tilde{\mathbb{U}}^k = \bigcup_i \tilde{\mathbb{U}}^{k,i}$, e si denoti con $\tilde{n}_x^k$, $\tilde{n}_w^k$, $\tilde{n}_u^k$ le cardinalità di questi insiemi.

Passo 1. Porre $k = k + 1$.

- Per $j = 1, \ldots, \tilde{n}_x^{k-1}$, selezionare i sottoinsiemi $\tilde{\mathbb{X}}^{k,j}$, $\tilde{\mathbb{W}}^{k,j}$, $\tilde{\mathbb{U}}^{k,j}$ che sono rilevanti per descrivere la dinamica dell'j-esimo elemento $\tilde{x}_t^{k,j}$ di $\tilde{\mathbb{X}}^{k-1}$, cioè tale che esista una funzione $\tilde{f}_t^j(\cdot)$ tale che $\tilde{x}_{t+1}^{k,j} = \tilde{f}_t^j(\tilde{\mathbf{x}}_t^{k,j}, \tilde{\mathbf{w}}_t^{k,j}, \tilde{\mathbf{u}}_t^{k,j})$.
- Sia $\tilde{\mathbb{X}}^k = \bigcup_j \tilde{\mathbb{X}}^{k,j}$, $\tilde{\mathbb{W}}^k = \bigcup_j \tilde{\mathbb{W}}^{k,j}$, $\tilde{\mathbb{U}}^k = \bigcup_j \tilde{\mathbb{U}}^{k,j}$.

Passo 2 (Test di terminazione). Se $\tilde{\mathbb{X}}^k \neq \tilde{\mathbb{X}}^{k-1}$, vai al Passo 3. Altrimenti, l'algoritmo termina e $\tilde{\mathbb{X}} = \bigcup_k \tilde{\mathbb{X}}^k$, $\tilde{\mathbb{W}} = \bigcup_k \tilde{\mathbb{W}}^k$, $\tilde{\mathbb{U}} = \bigcup_k \tilde{\mathbb{U}}^k$.

Passo 3. Porre $k = k + 1$.

- Si consideri l'insieme $\tilde{\mathbb{D}}^k = \tilde{\mathbb{X}}^{k-1} - \tilde{\mathbb{X}}^{k-2}$, la cui cardinalità è pari a $(\tilde{n}_x^{k-1} - \tilde{n}_x^{k-2})$.
- Per $j = 1, \ldots, (\tilde{n}_x^{k-1} - \tilde{n}_x^{k-2})$, selezionare i sottoinsiemi $\tilde{\mathbb{X}}^{k,j}$, $\tilde{\mathbb{W}}^{k,j}$, $\tilde{\mathbb{U}}^{k,j}$ che sono rilevanti per descrivere la dinamica dell'j-esimo elemento $\tilde{x}_t^{k,j}$ of $\tilde{\mathbb{D}}^k$, cioè tale che esista una funzione $\tilde{f}_t^j(\cdot)$ s.t. $\tilde{x}_{t+1}^{k,j} = \tilde{f}_t^j(\tilde{\mathbf{x}}_t^{k,j}, \tilde{\mathbf{w}}_t^{k,j}, \tilde{\mathbf{u}}_t^{k,j})$.
- Sia $\tilde{\mathbb{X}}^k = \bigcup_j \tilde{\mathbb{X}}^{k,j} \bigcup \tilde{\mathbb{X}}^{k-1}$, $\tilde{\mathbb{W}}^k = \bigcup_j \tilde{\mathbb{W}}^{k,j} \bigcup \tilde{\mathbb{W}}^{k-1}$, $\tilde{\mathbb{U}}^k = \bigcup_j \tilde{\mathbb{U}}^{k,j} \bigcup \tilde{\mathbb{U}}^{k-1}$.
- Ritornare al Passo 2.

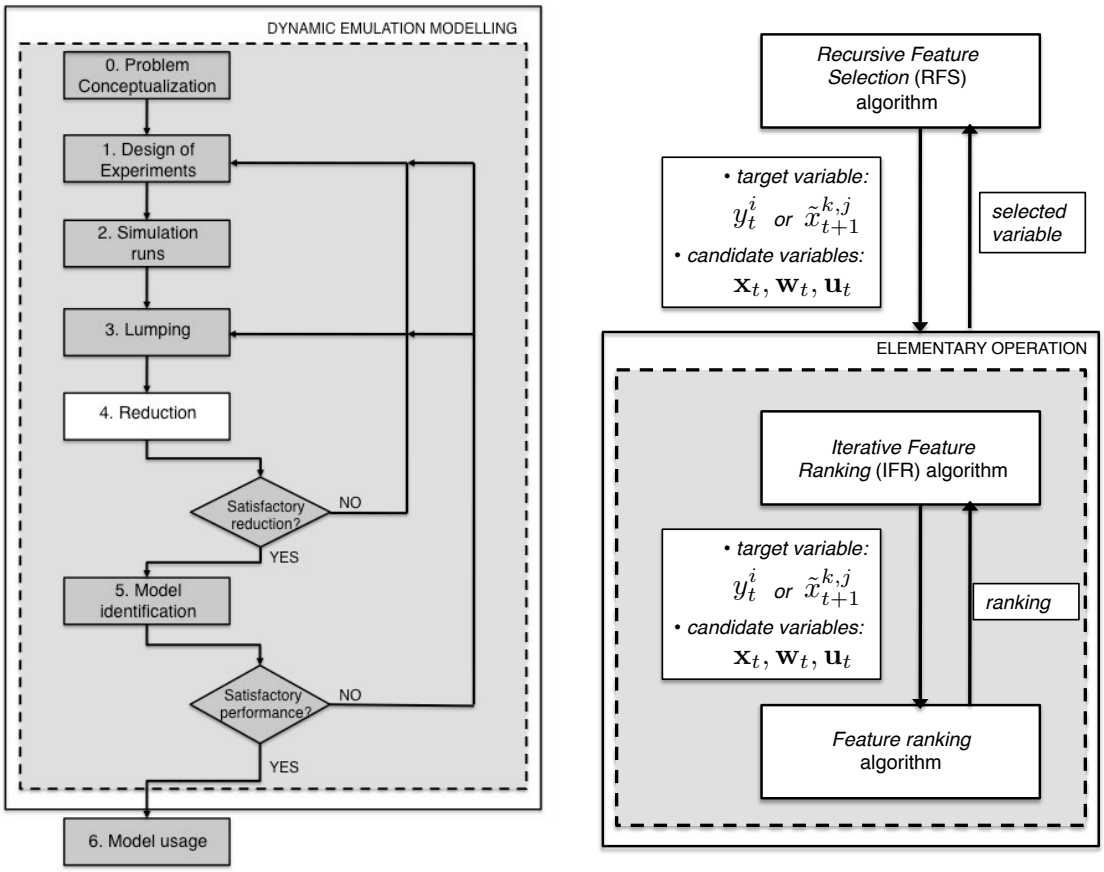

Figura 2.2: La procedura di *dynamic emulation modelling* e, a destra, un esploso della fase 4 (*riduzione*) concernente l'utilizzo degli algoritmi RFS e IFR (da Galelli, 2010)

L'approccio *model-free* si basa su misure statistiche che verificano la forza della relazione tra le features candidate e la variabile di output. Nella modellizzazione ambientale, le dipendenze da modellizzare sono in genere non lineari: per questo motivo, è più appropriato l'utilizzo di misure non lineari di dipendenza statistica, come la *Mutual Information* (MI). La MI è una misura di dipendenza non lineare tra due variabili, basata sull'approssimazione della loro funzione di densità di probabilità congiunta. Il suo maggiore svantaggio (Hejazi and Ximing Cai, 2009) consiste in una sua decrescente applicabilità nel caso di più di due variabili. Tale limite viene superato dall'algoritmo di Maximum Relevance (Stokelj *et al.*, 2002), che non tiene conto di alcuna ridondanza all'interno delle features, e perciò manca del potere discriminante di evitare la selezione di features ridondanti. La nuova frontiera per tenere conto sia della significatività sia dell'indipendenza delle features è l'algoritmo di *Minimum Redundancy - Maximum Relevance* (vedi Peng *et al.*, 2005; Hejazi e Ximing Cai, 2009). Il vantaggio principale dei metodi appartenenti all'approccio model-free è la loro velocità e i loro requisiti computazionali contenuti, ma essi richiedono un criterio arbitrario per interrompere la ricerca (ad esempio, quando viene selezionato un certo numero pre-specificato di features, oppure viene soddisfatto un certo criterio di soglia).

L'approccio *model-based* cerca all'interno dello spazio dei sottoinsiemi delle features utilizzando un modello per informare la ricerca. In particolare, calcola l'accuratezza del modello per ogni feature che può essere aggiunta o rimossa dal sottoinsieme delle features. Sono state sviluppate numerose tecniche per la ricerca: queste includono metodi ad hoc, dove lo sviluppatore del modello seleziona le combinazioni delle features che andrebbero provate; metodi step-wise, dove le features vengono sistematicamente aggiunte (forward-selection) o rimosse (backward-selection) dal subset delle features; metodi globali, dove viene utilizzato un algoritmo di ottimizzazione (ad esempio, algoritmi genetici) per selezionare le combinazioni delle features che massimizzano le prestazioni del modello (vedi, ad esempio, Bowden *et al.*, 2005). Un forte argomento a favore di questo approccio è che l'accuratezza stimata del modello utilizzato è la migliore informazione euristica disponibile per misurare l'importanza delle features candidate, mentre lo svantaggio principale è rappresentato dai suoi requisiti di calcolo, che crescono rapidamente all'aumentare del numero di features candidate.

2.9.2 L'algoritmo di Iterative Feature Ranking

L'algoritmo ideale da utilizzare all'interno dell'algoritmo RFS deve tener conto delle dipendenze non lineari e delle ridondanze tra le features, in quanto i processi simulati dai modelli PB ambientali sono tipicamente non lineari e fortemente correlati tra loro. Inoltre, deve essere computazionalmente efficiente, poiché il numero di features candidate è generalmente grande e il numero di chiamate dall'algoritmo RFS è rilevante. Per rispettare questi requisiti, è stato sviluppato (Galelli, 2010) un algoritmo di forward-selection totalmente model-free, denominato *Iterative Feature Ranking* (IFR).
Dato l'output r_t e il vettore $\mathbf{z}_t$ delle features candidate, l'algoritmo IFR innanzitutto classifica globalmente gli elementi di $\mathbf{z}_t$ in funzione di una misura statistica di significatività che tiene conto delle dipendenze non lineari, e in seguito raffina la classifica (o *ranking*) valutando il contributo individuale delle features classificate nelle prime p posizioni. La feature più significativa viene poi selezionata e utilizzata come regressore in un modello predefinito di classe $c(\cdot)$. Per tener conto della ridondanze delle features, l'algoritmo procede ripetendo queste operazioni su quei dati che devono ancora essere spiegati, cioè sui residui del modello costruito all'iterazione precedente. L'algoritmo itera queste operazioni finché la selezione di nuove features non migliora ulteriormente le prestazioni del modello in costruzione. L'algoritmo IFR richiede quindi di scegliere una misura statistica efficace, la quale, a sua volta, influenza la scelta della classe del modello $c(\cdot)$. Gli unici parametri da specificare sono perciò p e una tolleranza ε utilizzata per terminare l'algoritmo. I dettagli dell'algoritmo IFR sono mostrati nella Tabella 2.3.

Per quanto riguarda la classe del modello $c(\cdot)$ e la misura statistica, la proposta è quella di utilizzare una classe di metodi basati sugli alberi, chiamati *Extremely Randomized Trees* (Extra-Trees; Geurts *et al.*, 2006), e una procedura di ranking basata sugli Extra-Trees (per maggiori dettagli, vedi Appendice A).

Tabella 2.3: Algoritmo di Iterative Feature Ranking (da Galelli, 2010).

Passo 0. Porre $k = 0$ e $\tilde{\mathbf{z}}_t$ vettore nullo.

- Classificare, in ordine decrescente, le features all'interno del vettore $\mathbf{z}_t$ in funzione della loro misura statistica di significatività nello spiegare l'output r_t.

- Selezionare le features $z_t^1, \ldots, z_t^p$ classificate nelle prime p posizioni. Per $i = 1, \ldots, p$, identificare un modello della forma $\hat{r}_t^{k,i} = c(z_t^i)$ e valutare la sua prestazione R^i nello spiegare r_t.

- Denotare come z_t^k e $\hat{r}_t^k$ la feature e la stima di r_t corrispondente al modello con la prestazione più elevata R^k. Mettere da parte z_t^k all'interno di $\tilde{\mathbf{z}}_t$.

- Calcolare il residuo $e_t^k = r_t - \hat{r}_t^k$.

Passo 1. Porre $k = k + 1$.

- Classificare, in ordine decrescente, le features all'interno del vettore $\mathbf{z}_t$ in funzione della loro misura statistica di significatività nello spiegare l'output e_t^{k-1}.

- Selezionare le features $z_t^1, \ldots, z_t^p$ classificate nelle prime p posizioni. Per $i = 1, \ldots, p$, identificare un modello della forma $\hat{e}_t^{k-1,i} = c(z_t^i)$ e valutare la sua prestazione R^i nello spiegare e_t^{k-1}.

- Denotare come z_t^k la feature corrispondente al modello con la prestazione più elevata. Mettere da parte z_t^k all'interno di $\tilde{\mathbf{z}}_t$.

- Identificare un modello della forma $\hat{r}_t^k = c(\tilde{\mathbf{z}}_t)$ e valutare la sua prestazione R^k nello spiegare r_t.

- Calcolare il residuo $e_t^k = r_t - \hat{r}_t^k$.

Passo 2 (test di terminazione). Se $(R^k - R^{k-1}) < \epsilon$, l'algoritmo termina. Le features selezionate sono salvate all'interno di $\tilde{\mathbf{z}}_t$, con dimensionalità $\tilde{n}_z = k - 1$. Altrimenti, ritornare al Passo 1.

Capitolo 3

Il sistema della diga di Tono

Nel precedente capitolo 2, è stato proposto un approccio procedurale (Galelli, 2010) per l'identificazione di un *dynamic emulation model*. Lo scopo di questo lavoro di tesi è applicare la procedura ad un caso reale: il sistema della diga di Tono (Giappone). Il presente capitolo descrive le caratteristiche del sistema in esame, individua i criteri e i rispettivi costi per passo/indicatori, e sviluppa le possibili strategie per la risoluzione del problema di controllo ottimo.

3.1 Descrizione del sistema

La diga di Tono è in costruzione in questi anni alla confluenza dei fiumi Kango e Fukuro, e sarà completata nel 2012. Con un altezza di 75 m (ad una quota di 200 m s.l.m.), formerà un serbatoio artificiale di $12.4 \cdot 10^6$ m^3 (capacità di massima), con una superficie di 0.64 km^2 e alimentato da un bacino di 38.1 km^2.
Il serbatoio verrà costruito per soddisfare diversi obiettivi. In primo luogo, fornirà acqua per l'irrigazione di alcuni distretti agricoli a valle (per un'area irrigata totale di 353 ha), alimenterà una centrale idroelettrica da 1.1 MW, fornirà una riserva d'acqua per uso industriale pari a $30 \cdot 10^3$ m^3/giorno e una riserva d'acqua potabile per le comunità locali pari a $20 \cdot 10^3$ m^3/giorno; verrà inoltre usato per laminare l'effetto delle piene (fino a $5.5 \cdot 10^3$ m^3), per fornire servizi ecologici (es. habitat dei pesci) nel fiume a valle[1] e infine per scopi ricreativi. In accordo con l'approccio tradizionale spesso utilizzato nella pianificazione dei serbatoi, l'invaso totale di progetto è diviso in tre comparti principali, o strati (Fig. 3.1, che verranno usati per

[1] In base alla legge giapponese, i pesci all'interno del serbatoio non devono essere considerati.

scopi differenti: sul fondo, lo strato dei sedimenti, nel mezzo lo strato 'attivo', cioè quello utilizzato per soddisfare le richieste degli utilizzatori a valle, ed in cima un ulteriore strato per laminare le piene.
È stata progettata una struttura di rilascio selettivo (Selective Withdrawal Structure, o SWS, fig. 3.2), allo scopo di rilasciare l'acqua dello strato attivo a differenti livelli. Tale struttura è dotata di una serie di 15 sifoni disposti verticalmente a partire da 18 m dal fondo del serbatoio. I sifoni funzionano immettendo o togliendo aria, e il *blending* è permesso[2]. La quantità totale d'acqua rilasciata attraverso l'SWS è ugualmente suddivisa tra i sifoni aperti. L'acqua rilasciata attraverso l'SWS viene convogliata direttamente alla centrale idroelettrica. Un bypass sarà disponibile appena prima della centrale per allontanare qualsiasi portata più piccola o più grande di quella convogliabile alla centrale (rispettivamente 1.0 m^3/s e 3.0 m^3/s).
È prevista la realizzazione di uno scarico di piena ad un'altezza di 182.8 m (37.8 m dal fondo), esattamente sul fondo dello strato di laminazione delle piene. Il rilascio selettivo non è disponibile nello strato dei sedimenti, tuttavia è stato progettato uno scarico a 156 m per soddisfare il Deflusso Minimo Vitale (DMV) e supportare il rilascio in periodi estremamente secchi, nei quali il livello dell'acqua scende sotto il limite inferiore dello strato attivo. In condizioni normali il DMV è garantito dai sifoni, ma quando il livello scende al di sotto del limite inferiore dell'SWS, lo scarico nello strato dei sedimenti viene attivato. Questo scarico non può però essere utilizzato per scaricare i sedimenti.

3.2 Settori e criteri

Sebbene il principale scopo del serbatoio di Tono sia quello di fornire acqua per l'irrigazione agli agricoltori di valle, un certo numero stakeholders sarà affetto dalla costruzione della diga (vedi Castelletti *et al.*, 2010b). In questo paragrafo verranno analizzati tutti i settori, e per ognuno di essi verranno identificati i criteri che si suppone[3] possano essere adottati nel giudicare il livello di soddisfazione prodotto dai modi alternativi di operare l'SWS (cioè tramite differenti politiche di controllo),

[2]Per *blending* s'intende la possibilità di miscelare portate idriche con valori di temperatura differenti, attraverso l'utilizzo di più sifoni contemporaneamente: in questo modo, la portata in uscita dall'SWS avrà una temperatura intermedia tra quelle delle portate rilasciate attraverso i sifoni utilizzati.

[3]Poiché non è stata possibile un'interazione diretta con gli stakeholders, l'analisi di Castelletti *et al.*, 2010 è stata condotta in collaborazione con il Prof. H.Yajima dell'Università di Tottori.

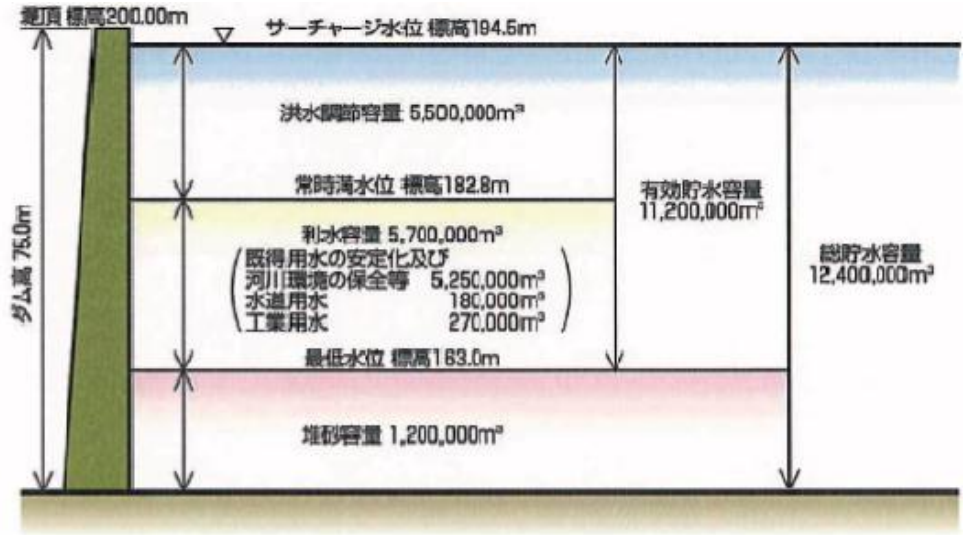

Figura 3.1: Schematizzazione dell'invaso della diga di Tono.

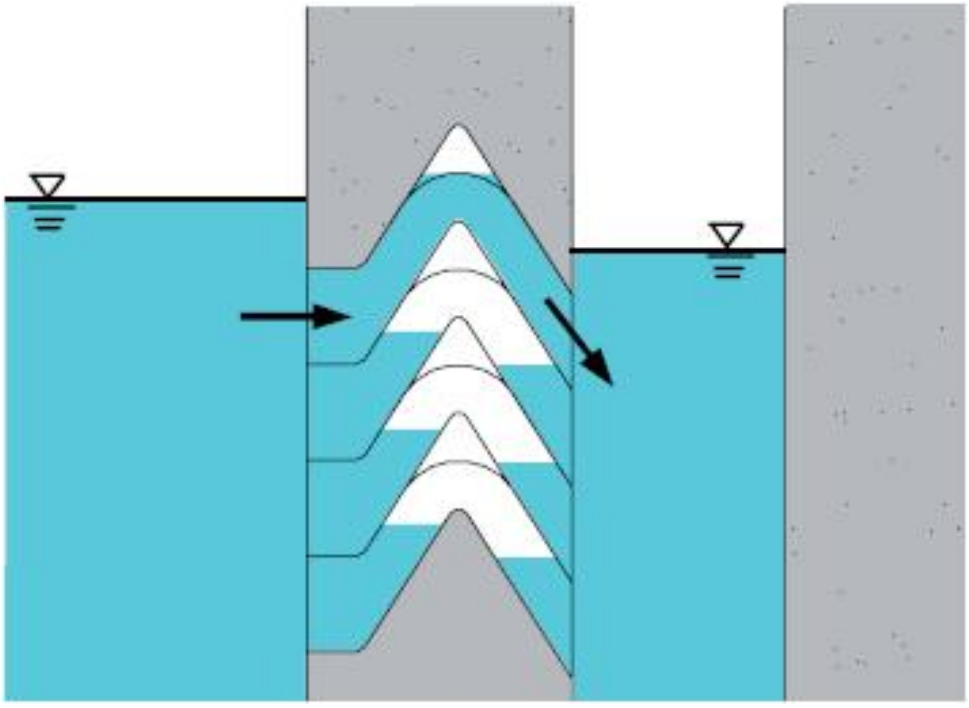

Figura 3.2: Schema della struttura di rilascio selettivo (SWS).

e gli effetti attesi prodotti dalla politica su questi criteri, che verranno validati a posteriori tramite simulazione della politica ottima ricavata dallo studio. Ai criteri (vedi Soncini-Sessa *et al.*, 2007) vengono associati indicatori quantitativi (o obiettivi) che permettono la comparazione delle politiche e la loro classificazione. Nel seguito, i differenti settori sono suddivisi in criteri di monte e di valle. Non tutti i settori verranno considerati nelle fasi successive del lavoro (vedi Tabella 3.1).

3.2.1 Criteri di monte

Il serbatoio verrà utilizzato per servizi ricreativi, e perciò il primo settore da considerare è il settore *Ricreazione*. Dall'altro lato, la *Sedimentazione* nel serbatoio gioca un ruolo chiave nella vita del lago, e deve essere tenuta al minimo per evitare un *silting*[4] troppo rapido dell'impianto.
Il settore *Ricreazione* è influenzato dalle modificazioni dell'aspetto del lago dovute alla crescita algale o da un alto livello di torbidità dell'acqua. Il controllo del profilo della temperatura è un metodo meccanico per controllare la profondità di intrusione del carico di nutrienti, e di conseguenza la distribuzione delle alghe, che è dipendente dalla luce disponibile. Il profilo di temperatura può variare di conseguenza al rilascio a differenti livelli. In generale, più in profondità avviene il rilascio, più in profondità scende il termoclinio. Questo implica, tuttavia, un rilascio di acqua più fredda con potenziali effetti negativi a valle.
La *Sedimentazione* dipende dall'afflusso e la risospensione può essere trascurata, data la profondità del serbatoio in progetto. Per poter rilasciare la maggior quantità di sedimenti possibile, l'acqua più torbida dovrebbe andare direttamente allo sfioratore, il quale è considerevolmente più largo dei sifoni dell'SWS. Tuttavia, l'intrusione dell'afflusso è governata dal profilo di temperatura all'interno del serbatoio e dalla temperatura dell'afflusso. Per questo motivo, per massimizzare il volume di sedimenti rilasciato, il rilascio dovrebbe avvenire alla profondità alla quale l'afflusso torbido sta penetrando. A queste latitudini, gli afflussi generalmente penetrano all'interno del corpo del lago appena sopra il termoclinio. Questo può suggerire il fatto che rilasciare dal sifone corrispondente allo strato superiore possa servire ad aumentare il lavaggio dei sedimenti. Inoltre, alcuni studi recenti hanno dimostrato che l'utilizzo del sifone più alto combinato con lo sfioratore porta l'afflusso alla profondità minima possibile e facilita il lavaggio dei sedimenti dallo sfioratore (Yajima

[4]Per *silting* s'intende il deposito e l'accumulo di sedimenti sul fondo del serbatoio, con conseguente aumento del livello dell'acqua.

et al., 2006). Questo modo di operare l'SWS può però avere effetti negativi su altri settori, come ad esempio l'irrigazione e l'ambiente di valle, che potrebbero essere danneggiati da acqua troppo calda. Anche il settore *Ricreazione* può essere danneggiato, in quanto tenendo il termoclinio nello strato meno profondo, la fioritura algale è favorita.

3.2.2 Criteri di valle

Idroelettrico, *Irrigazione*, *Temperatura*, *Ambiente* e *Piene* sono i settori identificati a valle della diga.
Il settore *Idroelettrico* è interessato a massimizzare il reddito dalla produzione elettrica, e in generale non è influenzato dalla qualità dell'acqua che passa attraverso le turbine. La produzione idroelettrica è considerata un obiettivo secondario, e come tale non sarà considerata nella fase di ottimizzazione.
La gerarchia di criteri per il settore *Irrigazione* è la stessa discussa in Soncini-Sessa *et al.* (2007, Ch.4), leggermente modificata per tenere conto degli effetti della temperatura sul raccolto. Infatti, si può dimostrare che la temperatura dell'acqua è in grado di influenzare le piante irrigate, specialmente nella fase di germogliazione. Acqua troppo fredda o troppo calda può ridurre il tasso di germogliazione e produrre foglie più piccole (perciò minore biomassa) in molte piante, mentre mantenere i valori di temperatura all'interno di un range appropriato può favorire la crescita. Il modo più semplice e più sensato dal punto di vista fisico per ridurre l'effetto delle variazioni di temperatura indotte, è quello di mantenere la temperatura del rilascio più simile possibile alla temperatura naturale dell'afflusso. Tutto ciò è importante anche dal punto di vista dei servizi ecologici forniti dalla diga, e dunque questo aspetto verrà considerato all'interno un settore separato chiamato *Temperatura*, che include sia gli agricoltori sia parte dell'ambiente. Oltre alla temperatura, gli agricoltori sono interessati a ridurre il deficit giornaliero di fornitura idrica, poiché questo ha un effetto diretto sul raccolto e di conseguenza sui loro ricavi (Soncini-Sessa *et al.*, 2007).
Il settore *Ambiente* è principalmente legato alla preservazione delle popolazioni di pesci minacciate da variazioni di temperatura e torbidità dell'acqua. Quest'ultima in particolare può avere impatti piuttosto forti, poiché periodi prolungati di alte concentrazioni di solidi sospesi totali (Total Suspended Solids, TSS) possono causare l'ostruzione delle branchie (e di conseguenza morte), la riduzione della capacità dei pesci di catturare prede (a causa della visibilità ridotta) e la distruzione degli

habitat. Per quanto riguarda la temperatura, la legislazione giapponese suggerisce di mantenerla più vicina possibile alle condizioni naturali, ad esempio a valori prossimi alla temperatura dell'afflusso. Per questo motivo, questo aspetto di protezione ambientale è stato incluso nel settore *Temperatura*.
Le *Piene* sono il problema maggiore per la città di Tottori, localizzata nei pressi del fiume a valle. Il criterio di questo settore è chiaramente quello di minimizzare i danni provocati dalle piene. Tuttavia, lo spazio atto a laminare le piene è assunto largamente sufficiente ad evitare le piene a valle, perciò questo settore non verrà considerato nelle analisi successive.

3.3 Costi per passo e indicatori

In linea di principio, ad ognuno dei criteri di settore specificati nel paragrafo 3.2 devono essere associati uno o più indicatori quantitativi, attraverso i quali differenti politiche di gestione possono essere valutate e comparate tra loro. Tuttavia, come già spiegato nello stesso paragrafo, i settori *Idroelettrico* e *Piene* non devono essere considerati nelle analisi successive e per questo motivo non è stato associato ad essi alcun costo per passo. Inoltre, il costo per passo formulato per il settore *Ambiente* non è stato incluso nell'ottimizzazione a causa di mancanza di dati utili per il settaggio di tutti i parametri che compaiono nella sua formulazione, e anche per mancanza di tempo per rifare le analisi con un'opportuna riformulazione dell'indicatore.
La formulazione dell'indicatore dipende dalla disponibilità di informazioni specifiche sui processi fisici e chimici. L'algoritmo fitted Q-iteration adottato in questo studio (vedi Appendice B) per progettare le politiche richiede che tutti gli indicatori siano separabili nel tempo[5]. Gli indicatori verranno perciò espressi come l'aggregazione di costi per passo g_t su di un orizzonte di tempo predefinito.

3.3.1 Ricreazione

Per questo settore è necessario formulare un costo per passo che sia in qualche modo fortemente correlato alla crescita algale. Una buona scelta potrebbe essere quella di considerare la concentrazione di fosforo nell'epilimnio, ma questa è difficilmente

[5]Ovvero, che gli indicatori possano essere valutati separatamente ad ogni istante di tempo

misurabile[6]. La concentrazione media di Chl-a nell'epilimnio è una valida alternativa potenziale. Tuttavia, dal momento che la Chl-a agisce come un proxy della crescita algale, ciò che conta è il massimo valore nello strato superficiale. Ecco perchè si considera la concentrazione massima media giornaliera di Chl-a nello strato superficiale

$$g_t^{rec} = \frac{1}{24} \sum_{\tau=1}^{24} \max_{z_\tau \in z_E} (\text{Chla}_\tau(z_\tau))^\alpha \tag{3.1}$$

dove Chl-a è la concentrazione di clorofilla [g/m^3] nella τ-esima ora del giorno t (ad es. tra $[t-1)$ e t), z_τ è la profondità misurata rispetto alla superficie del lago, z_E è la profondità dello strato superficiale (E sta per strato eufotico), settato a 7 metri sotto la superficie dell'acqua, così come risulta dalle analisi sulla trasparenza, e α è un coefficiente di amplificazione che tiene conto dell'effetto di crescita algale associato. Poiché $\alpha = 1$, l'indicatore per passo è espresso in g/m^3. L'indicatore sarà dunque formulato come la concentrazione media giornaliera sul periodo di validazione (vedi par. 4.2):

$$J^{rec}(p) = \frac{1}{H^{val}} \sum_{t=1}^{H^{val}-1} g_t^{rec} \tag{3.2}$$

dove H^{val} è il numero di giorni del periodo di validazione (1990-1994, vedi 4.2).

3.3.2 Sedimentazione

Allo scopo di ridurre il *silting* del serbatoio e aumentare la sua vita attesa, è necessario ridurre la sedimentazione all'interno del serbatoio. Un buon modo per fare questo può essere quello di massimizzare il volume giornaliero di sedimenti espulsi tramite il rilascio. Possiamo quindi considerare come costo per passo il seguente

$$g_t^{sed} = TSS_t^{out} \tag{3.3}$$

dove TSS_t^{out} è la quantità di Solidi Sospesi Totali [g/giorno] all'interno del deflusso tra $[t-1)$ e t. Più precisamente, TSS_{t+1}^{out} può essere calcolato come

[6]Un'altra variabile potenziale è il valore medio di Ossigeno Disciolto nello strato di fondo del serbatoio. Massimizzare il valore dell'ossigeno disciolto nell'acqua riduce la possibilità di condizioni anossiche nello strato di fondo, con conseguente rilascio di azoto e aumento di crescita algale. Tuttavia, questo tiene conto solamente del carico di nutrienti disponibile negli strati più profondi del serbatoio, ma ignora il contributo di nutrienti che entrano con l'afflusso.

$$TSS_t^{out} = \sum_{i=1}^{n} tss_t^i r_t^i + tss_t^{spill} r_t^{spill} \tag{3.4}$$

dove r_t^i [m^3/giorno] è il volume d'acqua rilasciato dall'i-esimo sifone degli n disponibili nell'SWS, e tss_t^i è la concentrazione media di TSS [g/m^3] nello strato corrispondente (si assume così che ad ogni sifone corrisponda uno strato); tss_t^{spill} sarà la concentrazione media di TSS nello strato dello sfioratore, e r_t^{spill} sarà il rilascio dallo strato corrispondente (di nuovo, si assume solo uno strato per ogni rilascio non regolato). L'indicatore relativo a questo settore sarà perciò la massima concentrazione media giornaliera di sedimenti espulsi tramite il rilascio:

$$J^{sed}(p) = \frac{1}{H^{val}} \sum_{t=1}^{H^{val}-1} TSS_t^{out} \tag{3.5}$$

3.3.3 Irrigazione

Il settore *Irrigazione* è interessato alla minimizzazione del deficit idrico. Siccome l'impatto della crescita vegetale sul deficit può avere differenti effetti in funzione della fase vegetativa, e l'effetto del deficit idrico sullo stress del raccolto reale non è lineare (riflettendo così una certa avversione al rischio degli agricoltori), è stato utilizzato il seguente costo per passo

$$g_t^{irr} = \beta(t)\Big(\big(w_t - (r_t - q_t^{MEF})\big)^+\Big)^\gamma \tag{3.6}$$

dove w_t è la domanda idrica agricola ridisegnata (per maggiori dettagli, vedi Castelletti *et al.*, 2010b), r_t è il rilascio totale dalla diga (inclusi SWS e sfioratore), q_t^{MEF} è il Deflusso Minimo Vitale e $(\cdot)^+$ è un operatore matematico che restituisce solo valori positivi del deficit, altrimenti restituisce zero. $\beta(t)$ è un coefficiente tempo-variante che considera la diversa rilevanza del deficit in differenti periodi dell'anno (ad es. la sommersione del riso e le diverse fasi vegetative). γ è un parametro che tiene conto dell'avversione al rischio degli agricoltori. Per quanto riguarda il valore dei parametri, $\beta(t)$ è pari a $\beta(t) = 1$ dal 3 maggio all'1 giugno; $\beta(t) = 0,8$ dal 2 giugno al 4 settembre; $\beta(t) = 0,3$ dal 5 settembre al 2 maggio. γ è posto pari a 2. L'indicatore per questo settore è il deficit irriguo medio sul periodo di validazione:

$$J^{irr}(p) = \frac{1}{H^{val}} \sum_{t=1}^{H^{val}-1} \beta(t)\big(w_t - (r_t - q_t^{MEF})\big)^2 \tag{3.7}$$

3.3.4 Temperatura

Il settore *Temperatura* è stato creato per includere gli interessi comuni dei settori *Irrigazione* e *Ambiente*. Il costo per passo relativo a questo settore è la differenza quadratica tra la temperatura dell'afflusso e la temperatura del rilascio

$$g_t^{temp} = (T_t^{out} - T_t^{in})^2 \tag{3.8}$$

dove T_t^{in} è definito come

$$T_t^{in} = \frac{T_t^K a_t^K + T_t^F a_t^F}{a_t^K + a_t^F} \tag{3.9}$$

dove T^K e T^F sono le temperature medie [℃] dell'afflusso tra $[t-1)$ e t dai fiumi Kango e Fukuro rispettivamente, a_t^K e a_t^F sono gli afflussi corrispondenti, mentre T_t^{out} è la temperatura media nello stesso intervallo di tempo in una sezione appena a valle dell'apertura delle turbine. Assumendo trascurabili gli effetti delle turbine sulla temperatura, così come la variazione di temperatura lungo il corso del fiume dalla diga a quella particolare sezione, T_t^{out} è data da

$$T_t^{out} = \frac{\sum_{i=1}^n T_t^i r_t^i + T_t^{spill} r_t^{spill}}{\sum_{i=1}^n r_t^i + r_t^{spill}} \tag{3.10}$$

dove T_t^i è la temperatura media tra $[t-1)$ e t nello strato corrispondente all'i-esimo sifone controllato e T_t^{spill} è la temperatura media nello strato dello sfioratore. $J^{temp}(p)$ è definito come la differenza quadratica media di temperatura tra l'afflusso al Tono e il deflusso da Tono, su tutto il periodo di validazione, cioè

$$J^{temp}(p) = \frac{1}{H^{val}} \sum_{t=1}^{H^{val}-1} \left(T_t^{out} - T_t^{in}\right)^2 \tag{3.11}$$

3.4 Il modello dinamico DYRESM-CAEDYM

Avendo a che fare con obiettivi quantitativi e qualitativi, la modellizzazione accurata e affidabile delle dinamiche del sistema diviene un'operazione estremamente complessa. In generale, un modello 3D spazialmente distribuito dei processi idrodinamici ed ecologici che si verificano nel lago sarebbe la migliore delle opzioni. Tuttavia, ci sono due ragioni per non adottare questo tipo di modello:

Tabella 3.1: Schema dei settori e degli indicatori considerati nel presente lavoro

Settore	Considerato	Indicatore
Ricreazione	Sì	$J^{rec}(p)$ (eq. (3.2))
Sedimentazione	Sì	$J^{sed}(p)$ (eq. (3.5))
Idroelettrico	No	-
Irrigazione	Sì	$J^{irr}(p)$ (eq. (3.7))
Temperatura	Sì	$J^{temp}(p)$ (eq. (3.11))
Ambiente	No	-
Piene	No	-

- Il serbatoio è stato creato sbarrando due fiumi in una sezione piuttosto stretta del loro corso, perciò dominano i fenomeni longitudinali e verticali. Un modello 2D di qualità dell'acqua potrebbe essere sufficiente per descrivere accuratamente fenomeni come l'intrusione dell'afflusso e la sedimentazione;

- Il rapporto stimato simulazione/tempo reale di un modello 3D è di 1/30 giorni, e ciò lo rende totalmente inadatto per il supporto alle decisioni.

Poiché un modello 2D di Tono non è disponibile, la scelta finale è stata quella di utilizzare un modello 1D, ovvero DYRESM-CAEDYM (Centre for Water Research, University of Western Australia), composizione del modello idrodinamico DYRESM con il modello ecologico CAEDYM.
DYRESM (acronimo di DYnamic REServoir Simulation Model) è un modello idrodinamico monodimensionale usato per prevedere le distribuzioni verticali di temperatura, salinità e densità nei laghi e nei serbatoi, e si basa sull'assunzione di monodimensionalità; ossia che le variazioni delle grandezze nella direzione verticale giochino un ruolo più importante di quelle in direzione orizzontale. L'approssimazione monodimensionale è valida quando le forze che destabilizzano un corpo idrico (vento, raffreddamento della superficie o afflussi) non agiscono per un periodo di tempo eccessivamente lungo. La dinamica di numerosi laghi e serbatoi, considerando scale temporali più lunghe di quelle di eventi estremi come tempeste e piene, è ben descritta dall'utilizzo di tale approssimazione. Inoltre, ciò rende possibile la descrizione del lago secondo una serie di strati o *layers* orizzontali. Non esiste alcuna variazione laterale o longitudinale negli strati, e il profilo verticale di qualunque

grandezza si ottiene dai valori della grandezza stessa in ogni strato. In DYRESM, gli strati hanno differente spessore; nel momento in cui gli afflussi e i deflussi entrano e lasciano il serbatoio, gli strati interessati si espandono o si contraggono, e gli strati superiori si spostano in alto o in basso per far fronte alla variazione di volume. Il movimento verticale degli strati è accompagnato da una variazione di spessore, in quanto le aree superficiali degli strati cambiano con la posizione verticale in funzione della batimetria. Il modello DYRESM parametrizza i processi fisici più importanti che nel tempo provocano cambiamenti nelle distribuzioni di temperatura, salinità e densità all'interno del serbatoio: tali parametrizzazioni derivano da studi dettagliati sui processi (sia sul campo che in laboratorio). Il modello risultante è perciò unico, in quanto previsioni affidabili sono ottenute senza calibrare i parametri del modello. DYRESM può funzionare autonomamente nel caso di studi puramente idrodinamici, oppure può essere accoppiato con CAEDYM (acronimo di Computational Aquatic Ecosystem DYnamics Model) per studi riguardanti processi chimici e/o biologici, come in questo caso. CAEDYM è un modello acquatico ecologico progettato per essere facilmente connesso ad un qualsiasi modello idrodinamico, ed è in grado di rappresentare i cicli del carbonio, del fosforo, dell'azoto, del silicio e dell'ossigeno disciolto, oltre alle dinamiche di diverse classi di solidi sospesi inorganici e fitoplancton. Numerose altre variabili di stato o biologiche possono essere configurate a discrezione dell'utente. CAEDYM viene generalmente eseguito con lo stesso passo temporale del modello idrodinamico al quale è legato.
Con questo modello composto si perdono le dinamiche spaziali tra l'ingresso e l'uscita del serbatoio (che verrebbero modellizzate da un modello 2D o 3D), tuttavia il *real-to-run time ratio* scende considerevolmente ad un valore di 1/12275 giorni. Paragonato ai semplici modelli usati tradizionalmente per il progetto di politiche di gestione dei serbatoi con obiettivi quantitativi (vedi Soncini-Sessa *et al.*, 2007), questo modello tiene conto di un numero elevato di variabili di stato, il che rappresenta un problema per la sua inclusione all'interno di un framework di ottimizzazione.

3.5 La metodologia e il modello

Lo scopo principale di questo lavoro di tesi è testare strumenti per la riduzione di un modello *process-based*, al fine di poter rendere risolvibile il problema di controllo. In questa sezione saranno formalizzati i concetti di politica di gestione e di soluzione ottima, e verrà presentata la metodologia adottata per costruire politiche Pareto-

ottimali.

3.5.1 La politica di gestione

Esistono essenzialmente due modi per supportare il decisore umano che si occupa della regolazione di un serbatoio:

1. L'approccio tradizionale, basato sul fornire al decisore una sequenza di 365 decisioni di rilascio (curva di rilascio), una per ogni giorno dell'anno (assumendo di decidere una volta al giorno), da applicare in condizioni 'normali';

2. L'approccio derivato dall'analisi dei sistemi, basato sul concetto di politica di gestione. All'operatore viene fornita una sequenza di 365 leggi di controllo, nella forma di relazioni matematiche, che forniscono la decisione di rilascio u_t al tempo t in funzione delle condizioni attuali del sistema (stato).

Mentre con il primo approccio qualsiasi deviazione dalle condizioni normali non solo è difficilmente quantificabile, ma è anche difficilmente correggibile, il secondo è un metodo di controllo in anello chiuso, che considera automaticamente le condizioni attuali del sistema e adatta a queste la decisione di rilascio. In generale, quando il sistema considerato è affetto da incertezza e non linearità (come nel nostro caso), le prestazioni dell'approccio basato sulla politica di controllo superano di gran lunga quelle dell'approccio basato sulla curva di rilascio.
Date le condizioni correnti del sistema (ovvero lo stato) $\mathbf{X}_t$, la legge di controllo

$$\mathbf{u}_t = m_t\left(\mathbf{X}_t\right) \tag{3.12}$$

dove $\mathbf{u}_t$ è il vettore delle decisioni di rilascio $\mathbf{u}_t = |u_t^1, \ldots, u_t^n|$, fornisce il volume $\mathbf{u}_t^i, i = 1, \ldots, n$, da rilasciare nell'intervallo di tempo $[t, t+1)$ (ad esempio nelle prossime 24 ore nel caso in cui si consideri una politica giornaliera come nel nostro caso) da ognuno degli n sifoni dell'SWS e per ogni giorno t dell'orizzonte di progetto[7]. Una politica di gestione p è definita come una sequenza temporale $p = \{m_0(\cdot), m_1(\cdot), \ldots\}$ di leggi di controllo della forma 3.12.

[7]Nel simbolo di una variabile, il pedice temporale denota l'istante di tempo in cui la variabile assume un valore deterministico, ad esempio l'invaso del lago è misurato al tempo t ed e quindi identificato con la dicitura s_t, mentre l'afflusso nell'intervallo $[t, t+1)$ è denotato con la dicitura a_{t+1} in quanto il suo valore è deterministicamente noto solo alla fine dell'intervallo. Nel caso delle funzioni, il loro pedice denota l'istante temporale, o l'istante iniziale dell'intervallo di tempo, alle quali si riferiscono.

3.5.2 Progetto della politica ottima

Formulazione del problema

Il problema di progetto di una politica di gestione per un serbatoio con SWS può essere formalizzato come un problema di ottimizzazione (più precisamente un problema di controllo ottimo) di un sistema dinamico affetto da un disturbo deterministico $\mathbf{W}_t$ (ad esempio gli afflussi, la radiazione solare, il carico di nutrienti presente nell'afflusso), nel quale la funzione obiettivo $J(p)$ da ottimizzare (massimizzare o minimizzare) esprime la soddisfazione del q-esimo criterio di settore considerato su tutto l'orizzonte H di progetto. Avendo a che fare con settori multipli e potenzialmente in conflitto tra loro, il concetto di politica ottima deve essere sostituito da quello di politica Pareto-ottimale (vedi Soncini-Sessa *et al.*, 2007).
Il modello del sistema è rappresentato in forma compatta dall'equazione vettoriale (1.2a), dove il vettore di stato $\mathbf{X}_t \in \mathcal{S}_{X_t} \subset \mathbb{R}^{N_x}$ include N_x variabili di stato; $\mathbf{u}_t \in \mathcal{U}_t(\mathbf{X}_t) \subseteq \mathbb{R}^{N_u}$, dove $\mathcal{U}_t(\mathbf{X}_t)$ è il vettore delle decisioni ammissibili, che dipendono dallo stato $\mathbf{X}_t$; il disturbo deterministico $\mathbf{W}_t \in \mathcal{S}_{W_t} \subseteq \mathbb{R}^{N_w}$ include N_w elementi.
Le uscite $\mathbf{Y}_t$ del modello, rappresentate dall'eq. (1.2b), sono i quattro costi per passo definiti nel paragrafo 3.3, e si riefriscono ai settori *Ricreazione* (eq. (3.1)), *Sedimentazione* (eq. (3.3)), *Irrigazione* (eq. (3.6)) e *Temperatura* (eq. (3.8)).

Sono disponibili diversi approcci per risolvere il problema (vedi Soncini-Sessa *et al.*, 2007), in grado di bilanciare in maniera differente requisiti computazionali e accuratezza della politica progettata. La Programmazione Dinamica Stocastica (PDS) è uno dei metodi più affidabili per progettare politiche di gestione Pareto-ottimali di sistemi di serbatoi. Sebbene sia stata studiata in maniera estensiva nella letteratura, la PDS soffre di una duplice problematica che di fatto impedisce la sua applicazione pratica a sistemi idrici complessi:

1. *Curse of Dimensionality*: i tempi di calcolo crescono esponenzialmente con le dimensioni dello stato, dei controlli e dei disturbi (Bellman, 1957), di modo che la PDS non può essere utilizzata per sistemi idrici dove il numero di serbatoi è più elevato di poche unità (2-3 serbatoi);

2. *Curse of Modelling*: è richiesto un modello esplicito di ogni componente del sistema idrico (Bertsekas e Tsitsiklis, 1996) per anticipare gli effetti delle tran-

sizioni del sistema, ovvero per descrivere ognuna delle transizioni di stato e i rispettivi costi associati.

Qualsiasi informazione utilizzabile dalla SDP può essere o una variabile di stato descritta da un modello dinamico, oppure un disturbo stocastico, bianco, descritto dalla sua funzione di densità di probabilità. L'informazione esogena (come la temperatura, la precipitazione o i volumi in afflusso al serbatoio) il cui utilizzo potrebbe migliorare notevolmente la gestione del serbatoio (Tejada-Guibert *et al.*, 1995; Hejazi *et al.*, 2008), non può essere considerata esplicitamente, a meno che non si introduca un modello dinamico per ognuna delle informazioni aggiuntive di cui si dispone, in altre parole trasformando le variabili esogene in variabili di stato, andando così ad aumentare il contributo della *Curse of Dimensionality*.
La *Curse of Modelling* della PDS ha ricevuto minore attenzione rispetto alla *Curse of Dimensionality*. Nella PDS, i modelli sono richiesti per prevedere e valutare gli effetti di ogni decisione plausibile sulle dinamiche dello stato, calcolandone il costo associato. Un approccio alternativo per effettuare tale valutazione è affidarsi direttamente all'esperienza. Questa è proprio l'idea centrale dell'Apprendimento per Rinforzo o Reinforcement Learning (RL), una procedura molto conosciuta per il decision-making sequenziale (Barto e Sutton, 1998) che combina concetti dalla PDS, dall'approssimazione stocastica, e dall'approssimazione di funzioni. L'esperienza di apprendimento può essere acquisita in linea, sperimentando direttamente le decisioni sul sistema reale senza l'ausilio di alcun modello, o generata fuori linea, utilizzando un simulatore esterno al processo di ottimizzazione oppure una serie di osservazioni storiche. Mentre la prima opzione è chiaramente impraticabile sulle reti di serbatoi reali, l'apprendimento fuori linea è già stato sperimentato con successo nella gestione dei sistemi idrici (Castelletti *et al.*, 2001; Soncini-Sessa *et al.*, 2007). I metodi basati sull'RL sono in grado di alleviare in qualche modo anche la *Curse of Dimensionality*, in quanto lo spazio in cui si ricercano le decisioni ammissibili di rilascio non viene esplorato in maniera esaustiva ad ogni passo di iterazione.

Soluzione del problema

L'approccio utilizzato in questo studio per il progetto delle politiche di regolazione di Tono Dam, chiamato *fitted Q-iteration* (Ernst *et al.* 2005), combina i concetti di apprendimento fuori linea tipico del RL con l'approssimazione funzionale della funzione valore. L'algoritmo fitted Q-iteration non richiede una modellizzazione

esplicita del sistema, e può essere utilizzato per il progetto di una politica di gestione, che viene determinata unicamente dall'apprendimento dall'esperienza. In particolare, tale esperienza è rappresentata da un dataset finito $\mathcal{T}$ di tuple della forma $< t, \tilde{\mathbf{x}}_t, \mathbf{u}_t, \tilde{\mathbf{x}}_{t+1}, g_t >$, ovvero:

$$\mathcal{T} = \{< t, \tilde{\mathbf{x}}_t^l, \mathbf{u}_t^l, \tilde{\mathbf{x}}_{t+1}^l, g_t^l >, l = 1, \ldots, \#\mathcal{T}\} \tag{3.13}$$

dove $\tilde{\mathbf{x}}_t$ è il vettore di stato ridotto argomento dell'eq. (1.3a), $\mathbf{u}_t$ è il vettore del controllo, g_t^l sono i costi per passo identificati nel par. 3.3, e $\#\mathcal{T}$ è la cardinalità di $\mathcal{TS}$. Ogni tupla è un esempio di transizione di un passo della dinamica del sistema di stato ridotto. Il dataset $\mathcal{T}$ è la sola informazione richiesta per la determinazione della politica di gestione, indipendentemente dal modo in cui viene generata.

Nel caso dell'algoritmo fitted Q-iteration, la funzione di costo viene approssimata tramite l'impiego di regressori di tipo tree-based (Breiman *et al.* 1984), i quali offre un'elevata flessibilità di modellizzazione, ed un'elevata efficienza di calcolo, poiché non è richiesta alcuna stima parametrica per l'approssimazione della funzione valore ad ogni passo d'iterazione. Inoltre, il fitted Q-iteration processa l'informazione disponibile (ovvero il dataset $\mathcal{T}$) in modo batch, utilizzando simultaneamente tutta l'esperienza di apprendimento per aggiornare la funzione valore, alleviando così gli effetti della *Curse of Modelling*. Nonostante ciò, l'approccio risente ancora del limite posto dalla *Curse of Dimensionality*, ovvero la crescita esponenziale dei costi di calcolo in funzione della dimensione dello stato (per maggiori dettagli sull'algoritmo di fitted Q-iteration,vedi App. B). Allo stato attuale, quindi, l'algoritmo fitted Q-iteration permette di risolvere problemi con circa 10 variabili di stato. Dal momento che, nella presente applicazione, il numero di variabili di stato del modello DYRESM-CAEDYM supera di gran lunga il numero di variabili gestibili dall'algoritmo di controllo ottimo, è possibile sfruttare tecniche di *emulation modelling* per risolvere il problema. In particolare:

1. L'emulation model può essere utilizzato direttamente per la risoluzione del problema di controllo: si costruisce cioè un modello di minor ordine (che comprende solamente $\tilde{n}_x$ variabili di stato) che si suppone sia in grado di sostituire completamente il modello DYRESM-CAEDYM in qualsiasi attività riguardante l'utilizzo dell'algoritmo di ottimizzazione;

2. Vengono utilizzate solamente le indicazioni sugli stati più significativi emerse durante la costruzione dell'emulation model per risolvere il problema di controllo, usando i dati originariamente generati dal modello DYRESM-CAEDYM per la costruzione del dataset $\mathcal{T}$.

3. Alternativamente, la riduzione dello stato può essere effettuata scegliendo in maniera empirica le variabili di stato che si suppone siano significative nel descrivere la dinamica dell'output $\mathbf{Y}_t$ (approccio *expert-based*).

Nell'ambito di questo lavoro di tesi, si è deciso di adottare la seconda modalità operativa. In primo luogo, verranno utilizzati gli algoritmi di features selection proposti nel Cap. 2 allo scopo di ridurre la complessità del modello DYRESM-CAEDYM, giungendo così alla costruzione di un modello ridotto (Cap. 4). In seguito, le informazioni relative agli stati più significativi e ai controlli verranno estrapolate dalle indicazioni fornite dall'identificazione dell'emulation model, e gli argomenti del dataset $\mathcal{T}$ in input all'algoritmo di fitted Q-iteration verranno selezionati dai dati di partenza di DYRESM-CAEDYM. I risultati della fase di ottimizzazione (Cap. 5) verranno infine confrontati con quelli ottenuti adottando l'approccio *expert-based* (Castelletti *et al.*, 2010b).

Capitolo 4

Identificazione dell'emulation model

Allo scopo di alleviare l'onere computazionale di cui si è parlato alla fine del capitolo precedente, così da rendere trattabile il problema di controllo ottimo, è necessario semplificare la descrizione dinamica del sistema in esame.
Lo scopo di questo capitolo è applicare l'approccio procedurale per il *dynamic emulation modelling* (Cap. 2) per ridurre il modello PB (DYRESM-CAEDYM nel nostro caso) che descrive le condizioni idrodinamiche ed ecologiche del serbatoio della diga di Tono, in Giappone. L'obiettivo di questo esperimento di emulation modelling è quello di ridurre la dimensionalità di DYRESM-CAEDYM, in modo che l'emulation model possa essere utilizzato per identificare le variabili di stato più significative rispetto all'output $\mathbf{Y}_t$, allo scopo di ottenere il dataset $\mathcal{T}$ (vedi par. 3.5.2). Il capitolo è strutturato come segue: la prossima sezione descrive le variabili di output del modello PB che devono essere simulate; il paragrafo 4.2 mostra le fasi di *Design of Experiments* (DOE) e *Simulation Runs*, mentre i paragrafi 4.3 e 4.4 riguardano rispettivamente le fasi di *lumping* e di riduzione. Infine, nel paragrafo 4.5 viene identificato l'emulation model.

4.1 Concettualizzazione del problema

4.1.1 Il modello PB

Il modello idrodinamico-ecologico monodimensionale DYRESM-CAEDYM è stato utilizzato per simulare i cicli stagionali e le dinamiche all'interno del serbatoio della

diga di Tono (vedi par. 3.4 per maggiori dettagli).
Nella presente applicazione al serbatoio di Tono, i disturbi esogeni $\mathbf{W}_t$ del modello includono 22 tipologie di variabili esogene relative ai dati di afflusso[1] e 6 variabili metereologiche (vedi Tabelle 4.1 e 4.2), tutte non distribuite nello spazio: la dimensionalità N_w di $\mathbf{W}_t$ è perciò pari a 50.

4.1.2 Variabili di output del modello PB e dell'emulation model

I quattro indicatori per passo definiti nel par. 3.3 rappresentano gli output del modello DYRESM-CAEDYM. Il vettore $\tilde{\mathbf{y}}_t$ degli output dell'emulation model è perciò caratterizzato da una dimensionalità $\tilde{n}_y$ pari a 4, mentre il passo temporale è uguale a 24 ore. L'obiettivo dell'esperimento di emulation modelling è quello di ridurre il più possibile il numero delle variabili di stato coinvolte nella descrizione di queste quattro variabili.

4.2 DOE e Simulation Runs

Lo scopo della fase di DOE è quello di esplorare, tramite simulazione del modello PB, l'area più ampia possibile all'interno degli spazi $\mathcal{L}_{\mathbf{Y}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$ e $\mathcal{L}_{\mathbf{X}_t} \times \mathcal{L}_{\mathbf{W}_t} \times \mathcal{L}_{\mathbf{u}_t}$. A questo proposito, per ogni simulazione, è necessario specificare una traiettoria sull'intero orizzonte di simulazione H per tutte le variabili di input del modello PB, ovvero la forzante esogena $\mathbf{W}_t$ e il controllo $\mathbf{u}_t$.
Per quanto riguarda $\mathbf{W}_t$, sono disponibili i dati orari delle realizzazioni dei processi idrologici e metereologici nei periodi 1990/1994, 1995/1999 e 2000/2006, per un totale di 17 anni di dati su base oraria. In realtà, si è deciso di utilizzare solamente i periodi 1995/1999 e 2000/2006 come input del modello DYRESM-CAEDYM, per un totale di 12 anni. Il periodo 1990/1994 verrà utilizzato nella fase di validazione delle politiche.
Per quanto riguarda il controllo, invece, il vettore delle decisioni di rilascio è stato fin qui definito come un vettore composto da n elementi (cioè $\mathbf{u}_t = |u_t^i, \ldots, u_t^n|$)

[1] In realtà queste vengono separate per ognuno dei due afflussi (i fiumi Kango e Fukuro), per un totale di 44 variabili esogene. Le variabili relative al particolare afflusso verranno denotate, nel seguito della trattazione, con le sigle (K) o (F) a seconda che siano relative ai fiumi Kango e Fukuro rispettivamente

Tabella 4.1: Riepilogo dei disturbi esogeni relativi agli afflussi del modello DYRESM-CAEDYM (forzanti $\mathbf{W}_t$).

Variabile	Descrizione	U.M.
$Volume_t$	volume dell'afflusso	m^3/h
$Tempture_t$	temperatura dell'afflusso	°C
$Salinity_t$	salinità dell'afflusso	%
$NH_{4,t}$	concentrazione ammoniaca	mgN/l
$NO_{3,t}$	concentrazione nitrati	mgN/l
$PONL_t$	azoto organico particolato	mgN/l
$PO_{4,t}$	concentrazione fosfati	mgP/l
$POPL_t$	fosforo organico particolato	mgP/l
DO_t	ossigeno disciolto	mgO/l
$DOCL_t$	carbonio organico disciolto	mgC/l
$POCL_t$	carbonio organico particolato	mgC/l
$SSOL1_t$	solidi sospesi inorganici (gruppo 1)	mg/l
$SSOL2_t$	solidi sospesi inorganici (gruppo 2)	mg/l
$SSOL3_t$	solidi sospesi inorganici (gruppo 3)	mg/l
$SSOL4_t$	solidi sospesi inorganici (gruppo 4)	mg/l
$SSOL5_t$	solidi sospesi inorganici (gruppo 5)	mg/l
$SSOL6_t$	solidi sospesi inorganici (gruppo 6)	mg/l
pH_t	pH	-
$SiO_{2,t}$	concentrazione silicati	mgSi/l
$CYANO_t$	concentrazione biomassa cianoficee	$\mu gChla/l$
$CHLOR_t$	concentrazione biomassa clorofite	$\mu gChla/l$
$FDIAT_t$	concentrazione biomassa diatomee	$\mu gChla/l$

Tabella 4.2: Riepilogo dei disturbi esogeni meteo del modello DYRESM-CAEDYM (forzanti $\mathbf{W}_t$).

Variabile	Descrizione	U.M.
SW_t	irraggiamento solare (short wave)	W/m^2
$Cloud - Cover_t$	copertura nuvolosa	%
$Air - Temp_t$	temperatura dell'aria	°C
$Vap - Press_t$	pressione del vapore	mbar
$Wind - Speed_t$	velocità del vento	m/s
$Rain_t$	pioggia	m

corrispondenti a n differenti sifoni dell'SWS. La scelta più naturale sarebbe quella di considerare tutti e 15 i sifoni, inclusi l'ultimo disponibile a 18 m dal fondo e quello relativo allo strato dei sedimenti posto a 11 m dal fondo del serbatoio. Tuttavia, il tempo di calcolo richiesto per progettare una politica di cotrollo per tutti questi sifoni sarebbe molto elevato, perciò è stato necessario operare una semplificazione. Basandosi su lavori precedenti (Castelletti *et al.*, 2010b; Garbarini, 2009; Galli, 2010), le variabili di controllo più efficaci dal punto di vista degli indicatori definiti in precedenza sono le decisioni di rilascio u_t^{-3} e u_t^{-13}, che forniscono i volumi d'acqua da rilasciare tra t e $[t+1)$ rispettivamente dai sifoni a -3 m e a -13 m di profondità rispetto alla superficie del serbatoio. L'idea di fondo è che queste altezze d'acqua corrispondano, nelle condizioni medie del serbatoio, all'epilimnio e all'ipolimnio del serbatoio stratificato.

Le variabili decisionali sono definite su di un insieme di ammissibilità $\mathcal{U}_t(s_t)$ che tiene conto di quali sifoni sono disponibili dato l'invaso s_t, dei limiti fisici imposti dai sifoni e dell'ampiezza dell'apertura dell'SWS, oltre dell'idraulica dell'SWS stesso. Più precisamente, ogni sifone non può convogliare più di 7.353 m^3/s, mentre il massimo deflusso permesso dall'SWS è pari a 13.780 m^3/s. Il volume d'acqua rilasciato attraverso ogni sifone non può essere deciso liberamente, ma dipende dalla quantità totale rilasciata dall'SWS, che viene divisa equamente tra i sifoni aperti. Va detto che quando più di un sifone è aperto, ogni sifone non può operare alla sua capacità massima.

Poiché il livello del serbatoio può variare in conseguenza dell'afflusso, dell'evaporazione e dell'acqua che viene rilasciata, il volume r_{t+1} rilasciato dall'SWS alla fine dell'intervallo $[t, t+1)$ può non corrispondere alla decisione di rilascio presa al tem-

po t. Per esempio, a causa di un afflusso elevato, il livello può salire oltre il livello più basso degli sfioratori, e una certa quantità d'acqua può fuoriuscire in maniera incontrollata dallo sfioratore. Ancora, l'acqua rilasciata può far sì che il livello scenda al di sotto del sifone attivo, e di conseguenza la decisione di rilascio deve essere riallocata ad un altro sifone. Infine, il Deflusso Minimo Vitale (DMV) e il supporto del deflusso in periodi particolarmente secchi possono richiedere l'apertura del sifone corrispondente allo strato dei sedimenti. La funzione di rilascio implementata in DYRESM-CAEDYM considera tutti questi eventi. Più precisamente, implementa le seguenti regole:

- quando il livello tra t e $t+1$ si alza al di sopra del livello più basso degli sfioratori, questi vengono attivati, e la quantità d'acqua rilasciata r_{t+1}^{spill} è data dalla curva caratteristica dello sfioratore;

- quando la decisione u_t^{-13} non può essere implementata, in quanto il sifone corrispondente non è disponibile, questa viene riallocata al sifone inferiore, in caso contrario il rilascio effettivo r_{t+1}^{-13} e la decisione di rilascio corrispondono;

- quando la decisione u_t^{-3} non può essere implementata, questa viene riallocata al sifone inferiore, in caso contrario il rilascio effettivo r_{t+1}^{-3} e la decisione di rilascio corrispondono;

- quando il DMV e il supporto del deflusso in periodi particolarmente secchi non vengono rilasciati dai sifoni controllati o dali sifone di fondo, viene aperto il sifone corrispondente allo strato dei sedimenti (r_{t+1}^{sed}).

Le tuple sono state generate simulando il modello 1D DYRESM-CAEDYM sul periodo idro-meteorologico 1995/2006 per 100 differenti scenari della decisione di rilascio $\mathbf{u}_t$ generati in modo pseudo-random con lo scopo di esplorare in maniera più omogenea possibile lo spazio di stato. I dati così ottenuti sono compresi nel dataset $\mathcal{F} = \{\mathbf{X}_t, \mathbf{W}_t, \mathbf{u}_t, \mathbf{Y}_t, \mathbf{X}_{t+1}\}$, contenente 437800 tuple. Per quanto riguarda le dimensionalità al termine della fase di *Simulation Runs*, $N_x \simeq 10^3$, $N_w = 50$, $N_u = 2$ e $N_y = 4$.

4.3 Lumping

In accordo con la procedura di *dynamic emulation modelling* (vedi Capitolo 2), le features più rilevanti $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ (nel descrivere il comportamento delle 4 variabili di output) devono essere selezionate tra tutte le variabili appartenenti a $\mathbf{X}_t$, $\mathbf{W}_t$ e $\mathbf{u}_t$, in base al contenuto informativo del dataset $\mathcal{F}$. Questa operazione è ben posta dal punto di vista teorico, ma è infattibile dal punto di vista computazionale, sia utilizzando algoritmi di features selection sia con considerazioni fisicamente basate, a causa della grande dimensionalità del vettore di stato $\mathbf{X}_t$. Per questo motivo, lo scopo della fase di *lumping* è trasformare, tramite aggregazione spaziale o temporale, i vettori $\mathbf{X}_t$ e $\mathbf{W}_t$ in vettori di dimensione minore $\mathbf{x}_t$ (con dimensionalità $n_x \ll N_x$) e $\mathbf{w}_t$ rispettivamente. Per quanto riguarda $\mathbf{u}_t$ e $\mathbf{W}_t$, non è richiesta alcuna trasformazione, poiché queste variabili non sono spazialmente distribuite (vedi par. 4.1.1). Per quanto riguarda $\mathbf{X}_t$, il *lumping*, nel caso specifico di questo lavoro di tesi, non consiste in un'aggregazione spaziale, ma in una scelta di alcune variabili ritenute potenzialmente significative nel descrivere il comportamento dell'output $\mathbf{Y}_t$. Queste variabili sono riportate in Tabella 4.3. La dimensionalità n_x di $\mathbf{x}_t$ è perciò pari a 19. Nel presente lavoro, si è scelto di attuare una riduzione del numero di tuple contenute nel dataset $\mathcal{F}$: il numero elevato di queste, unito al settaggio dell'algoritmo utilizzato durante la fase di riduzione (vedi par. 4.4) renderebbero gli esperimenti troppo onerosi in termini di tempo. A questo scopo, si adotta una clusterizzazione in funzione del controllo, più precisamente in funzione della somma dei due controlli disponibili, u_t^{-3} e u_t^{-13}, eseguendo poi un'analisi di frequenza preliminare sui dati e togliendo un numero di campioni proporzionale alla frequenza di ciascuna classe del controllo stesso. Una volta individuate le classi, si può operare il campionamento, applicando il fattore di riduzione scelto (90%) agli elementi che ricadono in ogni classe, per ottenere la matrice campionata che potrà essere utilizzata nella fase di riduzione. In questo modo, le tuple sono state ridotte del 90% (da 437800 a 43780), in modo da ottenere il dataset $\mathcal{F} = \{\mathbf{w}_t, \mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}, \mathbf{Y}_t\}$ per la fase di riduzione.

4.4 Riduzione

Il problema di riduzione richiede di selezionare le features $\tilde{\mathbf{x}}_t$, $\tilde{\mathbf{w}}_t$ e $\tilde{\mathbf{u}}_t$ che costituiscono gli argomenti della funzione di trasformazione d'uscita $\tilde{\mathbf{h}}_t(\cdot)$ e della funzione

Tabella 4.3: Riepilogo delle variabili di stato di DYRESM-CAEDYM $\mathbf{x}_t$.

Variabile	Descrizione	U.M.
t_{modT}	tempo	giorni
h_t	livello	m
s_t	invaso	m^3
$Taff_t$	temperatura dell'afflusso	°C
$hTaff_t$	livello di intrusione dell'afflusso	m
$gateTaff_t$	strato corrispondente all'intrusione dell'afflusso	-
$TSSmax_t$	massimo valore di TSS sull'intera colonna d'acqua	g/m^3
$hTSSmax_t$	livello del lago con max concentrazione TSS	m
$gateTSSmax_t$	strato corrispondente a max concentrazione TSS	-
T_t^{-3}	temperatura a -3 m	°C
T_t^{-7}	temperatura a -7 m	°C
T_t^{-13}	temperatura a -13 m	°C
T_t^{bot}	temperatura a 18 m dal fondo	°C
T_t^{sed}	temperatura a 11 m dal fondo (strato sedimenti)	°C
TSS_t^{-3}	concentrazione TSS a -3 m	g/m^3
TSS_t^{-7}	concentrazione TSS a -7 m	g/m^3
TSS_t^{-13}	concentrazione TSS a -13 m	g/m^3
TSS_t^{bot}	concentrazione TSS a 18 m dal fondo	g/m^3
TSS_t^{sed}	concentrazione TSS a 11 m dal fondo (strato sedimenti)	g/m^3

di transizione di stato $\tilde{\mathbf{f}}_t(\cdot)$. In particolare, l'algoritmo di *Recursive Feature Selection* (RFS, vedi par. 2.8) propone di selezionare in primo luogo le features rilevanti rispetto alla variabile di output $\mathbf{Y}_t$, ed in seguito di selezionare in maniera ricorsiva tutte le features rilevanti nello spiegare la dinamica degli stati $\tilde{\mathbf{x}}_{t+1}$. In questa applicazione, gli output sono costituiti dai costi per passo g_t^{temp}, g_t^{rec}, g_t^{sed} e g_t^{irr}, che rappresentano rispettivamente la differenza di temperatura dell'acqua in ingresso ein uscita dal serbatoio, la concentrazione di clorofilla all'interno del serbatoio, la concentrazione di sedimenti espulsi tramite il rilascio e il deficit irriguo (vedi par. 3.3).
Ogni operazione elementare dell'algoritmo RFS si basa sul'utilizzo dell'algoritmo di Iterative Feature Selection (IFR, vedi par. 2.9). Per ognuno degli output del modello, la tolleranza dell'algoritmo IFR ε è settata[2] a 0, ed il numero p di features valutate singolarmente è pari a 5. Per quanto riguarda la classe di modello $c(\cdot)$, vengono utilizzati gli Extra-Trees (vedi App. A). I loro parametri sono stati settati seguendo le indicazioni di Geurts *et al.* (2006): il numero di direzioni di taglio alternative K è 75 (ossia il numero di *candidate features*), pari al numero dei regressori; la minima cardinalità per suddividere un nodo n_{min} è pari a 50; il numero M di alberi che compongono la foresta è pari a 500, che rappresenta un buon compromesso tra l'accuratezza degli Extra-Trees e le richieste computazionali. Si ricorda inoltre che la fase di riduzione non viene eseguita sull'intero campione $\mathcal{F}$, bensì su di una sua sottoparte (vedi par. 4.3), in modo da ridurre i tempi di calcolo.
Le prestazioni dei modelli SISO e MISO ottenuti per ciascuna delle variabili considerate nella fase di riduzione verranno misurate tramite il coefficiente R^2, così definito

$$R^2 = 1 - \frac{cov(Q_i - \hat{Q}_i)}{cov(Q_i)} \tag{4.1}$$

dove Q_i sono i dati osservati, e $\hat{Q}_i$ quelli misurati.

4.4.1 Temperatura di valle g_t^{temp}

Il primo passo dell'algoritmo RFS richiede di identificare quali features, tra quelle contenute nei vettori $\mathbf{x}_t$, $\mathbf{w}_{t+1}$ e $\mathbf{u}_t$, sono rilevanti nel descrivere g_t^{temp}. Questo pas-

[2]Ciò significa che quando la selezione di una nuova feature porta ad un decremento di R^2, il processo di selezione degli input termina. Il valore di R^2 viene valutato tramite cross-validazione k-fold (con $k = 10$).

saggio viene eseguito tramite l'algoritmo IFR.
I risultati ottenuti al passo 0 dell'IFR, che classifica l'importanza di tutte le features candidate nello spiegare il comportamento di g_t^{temp}, ed in seguito valuta singolarmente l'importanza delle prime cinque classificate, sono riportati nella Tabella 4.4 *Passo 0*. Tra le prime cinque variabili classificate compaiono due temperature (la temperatura dello strato dei sedimenti T_t^{sed}, che è una variabile di stato, e la temperatura dell'afflusso dal fiume Kango T_{t+1}^K), i due controlli u_t^{-3} e u_t^{-13}, e la variabile $hTaff_t$, definita come l'altezza in corrispondenza della quale l'afflusso penetra nel corpo idrico. Ciò che si nota sono i valori molto bassi relativi a punteggi, riduzioni di varianza e prestazioni dei corrispondenti modelli SISO: la variabile che si classifica in prima posizione, T_t^{sed}, è in grado di spiegare da sola solamente poco più del 5% della dinamica di g_t^{temp}, e la prestazione del suo modello SISO è addirittura inferiore a quella della quinta variabile classificata, T_{t+1}^K (quest'ultima sarà la prima variabile selezionata dall'algoritmo, vedi Tabella 4.5). Tale fenomeno si ripete nei passi successivi: le Tabelle 4.4 - *Passo 1*, 4.4 - *Passo 1.1* e 4.4 - *Passo 1.2* mostrano come il ranking rimanga sostanzialmente invariato rispetto al *Passo 0*, e come i punteggi e le prestazioni dei modelli SISO rimangano molto bassi. Le variabili selezionate dall'algoritmo sono, nell'ordine, $hTaff_t$, T_t^{sed} e $NH_{4,t+1}^F$, con una prestazione MISO rispettivamente pari a 12%, 20% e 28% (vedi Tabella 4.5).
La situazione varia leggermente nei passi 1.3 e 1.4. Si nota un aumento dei punteggi relativi ai controlli u_t^{-3} e u_t^{-13} rispetto ai punteggi delle altre variabili classificate nei primi 5 posti (12% contro 4% circa al *Passo 1.3*, 13.8% contro 5% al *Passo 1.4*). Inoltre, le prestazioni dei modelli SISO relativi ai controlli, pur essendo piuttosto basse, sono comunque positive e superiori a quelle delle altre variabili ai primi 5 posti, che hanno prestazioni SISO negative: questa caratteristica si spiega con il fatto che a questo punto la maggior parte della dinamica di g_t^{temp} sia già stata spiegata, e ciò che rimane non è altro che rumore. Nonostante questo, le prestazioni MISO relative ai controlli non sono così elevate da poter affermare con sicurezza che la maggior parte della dinamica sia stata spiegata: la Tabella 4.5 mostra come i controlli portino il valore del coefficiente R^2 rispettivamente a 33.7 e 63.6 % (u_t^{-13} rappresenta il contributo più importante finora registrato), ma i valori delle prestazioni SISO delle altre variabili in cima alla classifica evidenzia l'avvicinamento alla condizione di over-fitting.
Quest'ultima affermazione trova riscontro negli ultimi passi dell'algoritmo (Tabelle 4.4 - *Passo 1.5*, 4.4 - *Passo 1.6* e 4.4 - *Passo 1.7*), con punteggi di varianza spiegata

molto bassi (max 2%) e prestazioni dei modelli SISO negative (oltre alla ricomparsa di variabili già selezionate in precedenza nelle prime cinque posizioni del ranking, caratteristica non segnata all'interno delle tabelle). La variabile selezionata al *Passo 1.7* è $Cloud-Cover_{t+1}$: tuttavia, l'introduzione di questa nuova variabile all'interno del modello MISO porta ad un decremento di R^2 pari a -0.0018%: considerando che la tolleranza ε è settata pari a zero, a questo punto l'algoritmo IFR si ferma, e le features selezionate sono quelle riportate nella Tabella 4.5.
L'adozione di questa condizione di terminazione dell'algoritmo è ben supportata da alcuni commenti riguardanti i risultati ottenuti al *Passo 1.7*, che mostrano come tutte le features candidate non possono essere utilizzate per spiegare il residuo e_t^7, che può essere considerato alla stregua di un disturbo bianco. I commenti sono:

- le prestazioni di alcuni dei modelli SISO sono negative;
- la varianza del residuo e_t^7 da spiegare è molto bassa, se paragonata alla varianza iniziale.

Tabella 4.4: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a g_t^{temp}.

Passo 0

Variabile di output	g_t^{temp}
Varianza variabile di output	$8.3762 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	5.3329	446698	0.0297
u_t^{-3}	5.1085	427897	0.0057
u_t^{-13}	4.9112	411367	0.0057
$hTaff_t$	4.0181	336566	0.0408
T_t^K	3.9953	334655	0.1139

Passo 1

Variabile di output	$g_t^{temp} - \hat{r}_t^0$
Varianza variabile di output	$7.4235 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	6.7224	499038	0.0103
u_t^{-3}	6.5719	487865	0.0061
u_t^{-13}	6.3781	473478	0.0058
$hTaff_t$	5.2379	388836	0.0175
T_t^{-3}	2.1825	162017	-0.0025

Passo 1.1

Variabile di output	$g_t^{temp} - \hat{r}_t^1$
Varianza variabile di output	$7.3419 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	5.9096	433876	0.0115
u_t^{-13}	5.8698	430958	0.0054
u_t^{-3}	5.8384	428649	0.0053
$Taff_t$	1.9573	143702	0.0026
$t_{mod\ T}$	1.9118	140359	0.0091

Passo 1.2

Variabile di output	$g_t^{temp} - \hat{r}_t^2$
Varianza variabile di output	$6.6653 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	7.5981	506442	0.0054
u_t^{-13}	7.54425	502850	0.0055
$NH_{4,t}^K$	1.9296	128612	0.0209
T_t^{-3}	1.9100	127310	-0.0031
$NH_{4,t}^F$	1.6388	109232	0.0209

Passo 1.3

Variabile di output	$g_t^{temp} - \hat{r}_t^3$
Varianza variabile di output	$6.0411 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	12.3908	748544	0.0061
u_t^{-13}	12.1586	734514	0.0054
T_t^{-3}	3.9488	238553	-0.0034
T_t^{-7}	2.6586	160611	-0.0033
T_t^{bot}	1.7821	107659	-0.0056

Passo 1.4

Variabile di output	$g_t^{temp} - \hat{r}_t^4$
Varianza variabile di output	$5.5713 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	13.8558	771947	0.0149
T_t^{-3}	4.9894	277975	-0.0034
T_t^{-7}	3.3983	189331	-0.0036
T_t^{-13}	1.6902	94164.2	-0.0049
h_t	1.4946	83267.6	-0.0052

Passo 1.5

Variabile di output	$g_t^{temp} - \hat{r}_t^5$
Varianza variabile di output	$2.9831 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.2563	67308.6	-0.0032
h_t	2.2303	66531.8	-0.0045
s_t	1.9761	58949.1	-0.0035
$t_{mod\ T}$	1.9557	58341.6	0.0045
T_t^{-13}	1.6294	48607	-0.0059

Passo 1.6

Variabile di output	$g_t^{temp} - \hat{r}_t^6$
Varianza variabile di output	$2.9567 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.3422	69250.8	-0.0032
T_t^{bot}	2.2970	67916.1	-0.0046
s_t	1.7385	51402.2	-0.0026
T_t^{-13}	1.6853	49828.6	-0.0048
T_t^{-7}	1.5605	46138.4	-0.0019

Passo 1.7

Variabile di output	$g_t^{temp} - \hat{r}_t^7$
Varianza variabile di output	$2.8462 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.2323	63535.1	-0.0048
h_t	2.1576	61410	-0.0028
s_t	1.8751	53367.3	-0.0026
T_t^{-13}	1.6064	45720.7	-0.0043
$Cloud - Cover_t$	1.2895	36700.3	0.0005

Tabella 4.5: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di g_t^{temp}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	T_t^K	0.1137	-
2	$\mathbf{hTaff}_t$	0.1238	0.0101
3	$\mathbf{T}_t^{sed}$	0.2080	0.0842
4	$NH_{4,t}^F$	0.2840	0.0760
5	u_t^{-3}	0.3375	0.0535
6	u_t^{-13}	0.6360	0.2985
7	$t_{mod\ T}$	0.6406	0.0046
8	$\mathbf{T}_t^{-7}$	0.6534	0.0128
9	$Cloud - Cover_t$	0.6516	0.0018

Le features $hTaff_t$, T_t^{sed} e T_t^{-7} sono variabili di stato, e come tali, la loro dinamica deve essere spiegata. In altri termini, le features rilevanti nello spiegare la dinamica di $hTaff_{t+1}$, T_{t+1}^{sed} e T_{t+1}^{-7} devono essere selezionate, tramite l'algoritmo IFR, al secondo passo dell'algoritmo RFS.

Dinamica di T_{t+1}^{sed}

I risultati dell'algoritmo IFR applicato a T_{t+1}^{sed} sono riportati nella tabella 4.6. Si nota la forte dipendenza di T_{t+1}^{sed} dal suo termine autoregressivo T_t^{sed} (valore di R^2 pari ad 80%), così come dalle temperature degli strati a -7 m e -3 m di profondità (5% incremento R^2 la prima, trascurabile la seconda), dai controlli u_t^{-3} e u_t^{-13}, dal tempo $t_{mod\ T}$, dalle variabili di stato $hTaff_t$ (già presente nella descrizione della dinamica di g_t^{temp}) e $gateTaff_t$, e da alcune forzanti esogene, come la pioggia, l'ossigeno disciolto e la temperatura dell'afflusso, che forniscono un contributo piuttosto scarso (la maggior parte delle forzanti esogene vengono selezionate nel momento in cui ci si avvicina alla condizione di over-fitting). Le due temperature T_t^{-7} e T_t^{-3}, e le variabili $hTaff_t$ e $gateTaff_t$ sono variabili di stato, pertanto la loro dinamica deve essere spiegata.

Tabella 4.6: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di T_{t+1}^{sed}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{T}_t^{sed}$	0.8079	-
2	$\mathbf{T}_t^{-7}$	0.8586	0.0507
3	u_t^{-3}	0.8658	0.0072
4	$t_{mod\,T}$	0.8890	0.0232
5	u_t^{-13}	0.8989	0.0099
6	$\mathbf{hTaff}_t$	0.9053	0.0064
7	$Rain_t$	0.9080	0.0027
8	DO_t^K	0.9090	0.0010
9	$\mathbf{gateTaff}_t$	0.9099	0.0009
10	DO_t^F	0.9101	0.0002
11	T_t^F	0.9108	0.0007
12	$\mathbf{T}_t^{-3}$	0.9119	0.0011
13	$Vap - Press_t$	0.9118	0.0001

Dinamica di $gateTaff_{t+1}$

I risultati dell'algoritmo IFR applicato a $gateTaff_{t+1}$ sono riportati nella tabella 4.7. La variabile $gateTaff_t$ è una variabile che indica lo strato in cui si intrude l'afflusso, e i suoi valori rappresentano delle 'etichette', piuttosto che dati numerici veri e propri: la variabile può perciò assumere i valori -3, -7, -13, 11 o 18. Il suo contenuto informativo è molto simile a quello di $hTaff_t$ (vedi nel seguito): il ruolo di entrambe è quello di dare indicazioni sulla stratificazione del serbatoio. Nel nostro caso, si evidenzia la dipendenza dal termine autoregressivo, anche se in maniera molto minore (coefficiente R^2 pari al 38%) rispetto al caso precedente relativo a T_{t+1}^{sed} (e anche rispetto a tutte le altre variabili di stato analizzate). Vengono poi selezionate, nell'ordine, le variabili $Air - Temp_t$ (variabile importante per la descrizione della stratificazione), $t_{mod\,T}$, $\mathbf{h}_t$ e $Cloud - Cover_t$ (ritenuta un proxy della luce solare). L'algoritmo si ferma quando seleziona $hTaff_t$, che rappresenta più o meno la stessa informazione contenuta in $gateTaff_t$, e quindi non migliora le prestazioni del modello MISO. Il punteggio finale del modello MISO ha comunque

un punteggio abbastanza basso (60%), uno dei peggiori relativamente alla dinamica delle variabili di stato selezionate dall'algoritmo IFR: con tutta probabilità, questo è dovuto alla particolare informazione contenuta nei dati di $gateTaff_t$. Tra tutte le variabili selezionate nel descrivere la dinamica di $gateTaff_t$, l'unica variabile di stato è il livello h_t, la cui dinamica dovrà poi essere spiegata.

Tabella 4.7: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di $gateTaff_{t+1}$. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{gateTaff}_t$	0.3814	-
2	$Air - Temp_t$	0.4948	0.1134
3	$t_{mod\ T}$	0.5599	0.0651
4	$\mathbf{h}_t$	0.5944	0.0345
5	$Cloud - Cover_t$	0.6074	0.0130
13	$\mathbf{hTaff}_t$	0.6064	0.0010

Dinamica di $hTaff_{t+1}$

Per quanto riguarda $hTaff_{t+1}$, la lista delle variabili selezionate è riportata nella Tabella 4.8: si può notare come $hTaff_{t+1}$ dipenda fortemente dal suo termine autoregressivo $hTaff_t$ (da solo è in grado di spiegare circa il 50% della dinamica di $hTaff_{t+1}$), dal tempo (da solo spiega il 10% del residuo) e da alcune forzanti meteo ($Air - Temp_t$, $Vap - Press_t$ e $Cloud - Cover_t$), la più importante delle quali è la temperatura dell'aria, che influisce direttamente su rimescolamento all'interno del lago, e indirettamente sulla quota alla quale penetrerà l'afflusso. Si nota anche la dipendenza da T_t^{-7}, ovvero la temperatura dello strato a -7 m di profondità: questa dipendenza è 'curiosa', in quanto si poteva notare anche per quanto riguarda g_t^{temp}, anche in quel caso con valori molto bassi di varianza spiegata e prestazioni SISO (e di conseguenza aumenti minimi della prestazione del modello MISO). In ogni caso, T_t^{-7} è una variabile di stato, e richiede perciò la descrizione della sua dinamica.

Tabella 4.8: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di $hTaff_{t+1}$. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{hTaff}_t$	0.4992	-
2	$Air - Temp_t$	0.5725	0.0733
3	$Vap - Press_t$	0.6191	0.0466
4	$t_{mod\ T}$	0.7125	0.0934
5	$Cloud - Cover_t$	0.7270	0.0145
6	$\mathbf{T}_t^{-7}$	0.7499	0.0229
7	u_t^{-13}	0.7472	0.0027

Dinamica di T_{t+1}^{-7}

I risultati dell'algoritmo IFR applicato a T_{t+1}^{-7} sono riportati nella Tabella 4.9. Si nota subito l'assenza di dipendenza dal termine autoregressivo T_t^{-7}, in favore della dipendenza da T_t^{-3}. Questo fatto si può spiegare in tre modi:

1. i termini T_t^{-7} e T_t^{-3} hanno più o meno lo stesso contenuto informativo: ciò è chiaro osservando i risultati della prima iterazione dell'algoritmo IFR (Appendice C, Tabella C.8), in cui le prestazioni dei modelli SISO delle due variabili sono quasi identici (81.18% contro 81.04% del termine autoregressivo);

2. il profilo di temperatura dipende fortemente dall'invaso del serbatoio: nel momento in cui il livello scende al di sotto degli strati attivi, la temperatura di questi strati viene automaticamente allocata al primo strato disponibile (vedi Figura 4.1): ciò significa che, in tutti gli intervalli di tempo in cui il serbatoio viene svoutato, le temperature degli strati attivi vengono fuse tra loro in un unico valore, e questo spiega l'indifferenza dell'algoritmo IFR nello scegliere l'una o l'altra delle due temperature al passo 1;

3. dal punto di vista fisico, questo fenomeno è del tutto normale, in quanto la posizione del termoclinio, che coincide con lo strato a -7 m di profondità, dipende dalla stratificazione del lago, per cui la temperatura di uno strato sarà direttamente influenzata da quelle degli strati superiori ed inferiori. Le temperature degli strati sono perciò strettamente correlate tra loro.

L'affermazione al punto 3 trova riscontro dalla dipendenza di T_{t+1}^{-7} da T_t^{-13}, che rappresenta la temperatura dello strato direttamente inferiore. Si nota anche la dipendenza dai controlli u_t^{-3} e u_t^{-13}(dipendenza riscontrata ogni volta in cui si cerca di spiegare la dinamica di una temperatura o di una differenza di temperatura), da $hTaff_t$, da $NH_{4,t+1}$ di entrambi gli afflussi (quello relativo a Fukuro compare in ogni dinamica di temperatura o differenza di temperatura, quello relativo a Kango è pressoché identico al primo, e viene selezionato per ultimo, quando il residuo e_t da spiegare si può considerare un disturbo bianco), dal livello h_t e da altre forzanti esogene ($Vap{-}Press_t$, T_t^K e SW_t). Dalla lettura della Tabella 4.9 si può notare come T_t^{-3} fornisca il contributo maggiore (R^2= 81%), mentre le altre variabili sommate danno un contributo totale pari al 9%. In particolare, tutte le forzanti esogene selezionate dopo il livello h_t non aumentano la prestazione del modello MISO oltre il 90%.
Tra tutte queste features selezionate, T_t^{-3}, T_t^{-13} e h_t sono variabili di stato, e la loro dinamica deve perciò essere spiegata.

Tabella 4.9: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di T_{t+1}^{-7}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{T}_t^{-3}$	0.8118	-
2	$Vap - Press_t$	0.8304	0.0186
3	u_t^{-3}	0.8467	0.0163
4	u_t^{-13}	0.8614	0.0147
5	$\mathbf{T}_t^{-13}$	0.8761	0.0147
6	$\mathbf{hTaff}_t$	0.8802	0.0041
7	$NH_{4,t}^F$	0.8901	0.0099
8	$\mathbf{h}_t$	0.9030	0.0129
9	T_t^K	0.9061	0.0031
10	SW_t	0.9064	0.0003
11	$NH_{4,t}^K$	0.9071	0.007
12	$Cloud - Cover_t$	0.9070	0.0001

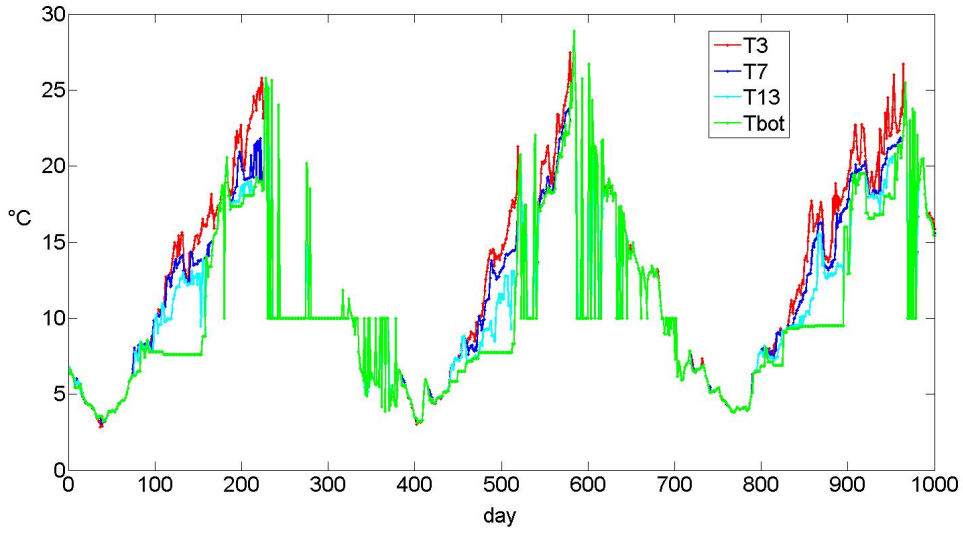

Figura 4.1: Temperature degli strati a -3, -7 e -13 m dalla superficie e a 18 m dal fondo (anni 1995-1997)

Dinamica di T_{t+1}^{-3}

I risultati dell'algoritmo IFR applicato a T_{t+1}^{-3} sono riportati nella Tabella 4.10. Si evidenzia subito la dipendenza dal termine autoregressivo (da solo ha un coefficiente R^2 pari ad 82.5%), dal livello, di nuovo dai controlli, dalla temperatura dell'aria, ancora da $NH_{4,t}^F$ (vedi sopra) ed infine da altre forzanti esogene dal contributo limitato ($Rain_t$, T_t^K e SW_t, sommate danno un contributo dello 0.3%).

Tabella 4.10: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di T_{t+1}^{-3}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{T}_t^{-3}$	0.8253	-
2	u_t^{-3}	0.8363	0.0110
3	$\mathbf{h}_t$	0.8716	0.0353
4	$Air - Temp_t$	0.8920	0.0204
5	u_t^{-13}	0.9113	0.0193
6	$NH_{4,t}^F$	0.9173	0.0060
7	$Rain_t$	0.9182	0.0009
8	T_t^K	0.9200	0.0018
9	SW_t	0.9203	0.0003
10	$Cloud - Cover_t$	0.9201	0.0002

Dinamica di T_{t+1}^{-13}

I risultati dell'algoritmo IFR applicato a T_{t+1}^{-13} sono riportati nella Tabella 4.11. Si nota, come nel caso precedente, la forte dipendenza dal termine autoregressivo (R^2 = 79%), dal livello, dai controlli, da $hTaff_t$ e da altre forzanti esogene più o meno ricorrenti e dal contributo limitato ($Vap - Press_t$, $Rain_t$, T_t^K, DO_t^F e T_t^F, insieme forniscono un contributo del 2% scarso). Non viene selezionata nessuna ulteriore variabile di stato.

Tabella 4.11: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di T_{t+1}^{-13}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{T}_t^{-13}$	0.7917	-
2	u_t^{-3}	0.8065	0.0148
3	u_t^{-13}	0.8203	0.0138
4	$\mathbf{hTaff}_t$	0.8424	0.0221
5	$\mathbf{h}_t$	0.8678	0.0254
6	$Vap - Press_t$	0.8812	0.0134
7	T_t^K	0.8843	0.0031
8	$Rain_t$	0.8858	0.0015
9	DO_t^F	0.8864	0.0006
10	T_t^F	0.8865	0.0001
11	DO_t^K	0.8865	0.0000

Dinamica di h_{t+1}

Infine, i risultati relativi ad h_{t+1} sono riportati nella tabella 4.12. Ciò che si nota è l'assoluta preponderanza del termine autoregressivo h_t (da solo spiega il 98% della varianza), la presenza dei controlli u_t^{-3} e u_t^{-13} e la dipendenza da SW_t, che può essere considerato un proxy dell'evaporazione. Di contro, si nota la presenza di variabili assolutamente non correlate al livello, come i solidi sospesi inorganici (addirittura 4 categorie, nonostante abbiano praticamente lo stesso contenuto informativo, di cui una viene selezionata prima dei controlli e della luce solare) ed $NO_{3,t}^F$: va detto però che il loro contributo relativamente alla dinamica di h_{t+1} è di fatto nullo. Colpisce anche la mancanza dei volumi in entrata dai fiumi Kango e Fukuro, due forzanti esogene fisicamente correlate al livello: si ipotizza che l'assenza degli afflussi tra le variabili selezionate sia dovuta al peso del termine autoregressivo h_t, il quale è talmente elevato che l'algoritmo fatica a distinguere l'importanza del contributo relativo ti tutte le altre variabili candidate. Ciò spiega anche la presenza di variabili fisicamente scorrelate al livello, come le categorie di solidi sospesi inorganici.

Tabella 4.12: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di h_{t+1}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$\mathbf{h}_t$	0.9820	-
2	$SSOL4_t^K$	0.9894	0.0074
3	u_t^{-13}	0.9939	0.0045
4	u_t^{-3}	0.9974	0.0035
5	SW_t	0.9975	0.0001
6	$NO_{3,t}^F$	0.9975	0.0000
7	$SSOL5_t^F$	0.9975	0.0000
8	$SSOL6_t^K$	0.9975	0.0000
9	$SSOL6_t^F$	0.9976	0.0001
11	$SSOL2_t^F$	0.9975	0.0001

A questo punto, il processo RFS può ritenersi concluso: per spiegare la dinamica di g_t^{temp}, sono state selezionate complessivamente sette variabili di stato (T_t^{sed}, $hTaff_t$, T_t^{-7}, T_t^{-3}, T_t^{-13}, $gateTaff_t$ ed h_t), il tempo $t_{mod\,T}$, dieci forzanti esogene ($NH_{4,t}$, T_t^K, T_t^F, DO_t^K, DO_t^F, SW_t, $Cloud-Cover_t$, $Vap-Press_t$, $Air-Temp_t$ e $Rain_t$) e i due controlli u_t^{-3} e u_t^{-13}.

4.4.2 Clorofilla nel serbatoio g_t^{rec}

L'algoritmo RFS viene ora applicato al caso della clorofilla all'interno del serbatoio, in altre parole per selezionare le features più rilevanti rispetto alla dinamica del costo per passo g_t^{rec} (ovvero della concentrazione media di clorofilla nel serbatoio). Così come nel caso della temperatura, le features sono selezionate tramite l'algoritmo IFR.

I risultati ottenuti sono mostrati in Tabella 4.13. La variabile g_t^{rec} dipende principalmente dalla concentrazione di ossigeno disciolto disponibile nell'afflusso (DO_t^K),

dalla concentrazione di nutrienti proveniente dall'afflusso ($NO^F_{3,t}$), dal livello del lago h_t (quindi dalla quantità d'acqua disponibile), dal tempo e dalla copertura nuvolosa $Cloud - Cover_t$ (proxy dell'irraggiamento solare). La concentrazione di ossigeno disciolto è chiaramente correlata alla clorofilla, nel senso che la disponibilità di ossigeno favorisce le esplosioni algali, le quali a loro volta provocano un decremento dell'ossigeno disponibile all'interno del corpo idrico. Anche la dipendenza dalla concentrazione di azoto è chiara, in quanto i nutrienti sono un fattore limitante, insieme ad ossigeno e luce solare, della fioritura algale. Ciò che può destare dubbi è l'assenza di una variabile che rappresenti l'apporto di fosforo al lago (come ad esempio $PO_{4,t+1}$: questo si spiega con il fatto che le concentrazioni di fosforo, pressoché identiche per entrambi gli afflussi, sono considerevolmente minori di quelle di azoto, e, tenendo conto che le portate dei due affluenti Kango e Fukuro sono molto simili, il carico di fosforo al serbatoio può essere ritenuto trascurabile rispetto a quello d'azoto. La dipendenza dal livello h_t può essere spiegata dal fatto che quando il serbatoio viene svuotato, o comunque il livello viene mantenuto ad un valore piuttosto basso, le fioriture algali diminuiscono (vedi Castelletti *et al.*, 2010b per maggiori dettagli). Colpisce la mancata dipendenza dai controlli: questo potrebbe sembrare un errore, alla luce del fatto che in periodi di fioriture algali, la strategia più utilizzata per contrabattere il problema è rilasciare il più possibile, ovvero agire sul controllo. In realtà, i controlli entrano a far parte indirettamente del modello di g^{rec}_t nella spiegazione della dinamica di h_t (par. 4.4.1, Tabella 4.12).

Tabella 4.13: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di g^{rec}_t. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	DO^K_t	0.6129	-
2	$NO^F_{3,t}$	0.7618	0.1489
3	$\mathbf{h}_t$	0.8245	0.0627
4	$t_{mod\ T}$	0.8937	0.0692
5	$Cloud - Cover_t$	0.8993	0.0056
6	$\mathbf{T}^{sed}_t$	0.8985	0.0008

Il processo RFS è concluso: per spiegare la dinamica di g^{rec}_t, sono state selezionate complessivamente una variabile di stato (h_t), il tempo $t_{mod\ T}$, sette forzanti esogene

($SSOL4_t$, $SSOL5_t$, $SSOL6_t$, DO_t^K, $NO_{3,t}^F$, SW_t e $Cloud-Cover_t$) e i due controlli u_t^{-3} e u_t^{-13}.

4.4.3 Sedimentazione nel serbatoio g_t^{sed}

Si consideri ora l'applicazione dell'algoritmo RFS (e dell'IFR per la selezione delle features) nel caso dell'output g_t^{sed}. I risultati sono mostrati in Tabella 4.14. Come ci si potrebbe aspettare, le dipendenze sono esclusivamente relative alle concentrazioni di sedimenti: in particolare, vengono selezionate 4 classi di solidi sospesi inorganici (SSOL2, SSOL3, SSOL4 e SSOL5, ricordando che le concentrazioni di solidi sospesi inorganici sono identiche per entrambi gli afflussi) e una variabile di stato, TSS_t^{-3}, che rappresenta la concentrazione di solidi sospesi totali nello strato a -3 m di profondità. Considerando che le uniche variabili di controllo utilizzate in questo studio sono quelle relative alle bocche a -3 m e a -13 m di profondità, ci si potrebbe aspettare la dipendenza anche da TSS_t^{-13}, cosa che non avviene. In realtà, la modellizzazione monodimensionale effettuata da DYRESM-CAEDYM fa in modo che le informazioni sul trasporto orizzontale dei sedimenti dagli afflussi alle bocche dell'SWS vadano perse: per questo motivo, è come se i sedimenti venissero scaricati nel momento in cui entrano all'interno del serbatoio, in corrispondenza dello strato superficiale, cioè lo strato a -3 m di profondità. Quest'ultima affermazione è ulteriormente supportata dall'assenza, tra le variabili selezionate, dei controlli u_t^{-3} e u_t^{-13}, da cui si evince che la strategia migliore per minimizzare la concentrazione dei sedimenti all'interno del serbatoio è scaricare il carico di sedimenti in entrata dagli affluenti non appena giungono all'imbocco dell'SWS.
L'assenza del controllo tra le variabili selezionate per spiegare la dinamica di g_t^{sed} mostra come il problema di controllo nel caso di questo costo per passo sia in realtà mal posto: per questo motivo, l'indicatore $J^{sed}(p)$ non verrà preso in considerazione nella successiva fase di ottimizzazione (Cap. 5).

Tabella 4.14: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di g_t^{sed}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$SSOL5_t^K$	0.4134	-
2	$SSOL3_t^F$	0.7010	0.2876
3	$SSOL4_t^F$	0.7631	0.0621
4	$\mathbf{TSS}_t^{-3}$	0.8405	0.0774
5	$SSOL2_t^K$	0.8938	0.0533
6	$SSOL4_t^K$	0.8870	0.0068

Dinamica di TSS_{t+1}^{-3}

Come anticipato, la variabile TSS_t^{-3} è una variabile di stato, e come tale, le features rilevanti nello spiegare la dinamica di TSS_{t+1}^{-3} devono essere selezionate, tramite l'algoritmo IFR, al passo successivo dell'algoritmo RFS. La lista delle features selezionate è riportata nella Tabella 4.15. Il risultato è simile a quello relativo alla dinamica di g_t^{sed}, e ciò non desta sorprese: si evidenzia la dipendenza dal termine autoregressivo TSS_t^{-3} e da alcune categorie di solidi sospesi inorganici (SSOL1 e SSOL2) e di forzanti esogene sempre relative a particolati (POCL e DOCL). Non vengono selezionate ulteriori variabili di stato.

Tabella 4.15: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di TSS_{t+1}^{-3}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	$SSOL1_t^F$	0.7668	-
2	$\mathbf{TSS}_t^{-3}$	0.8183	0.0515
3	$SSOL2_t^F$	0.8248	0.0065
4	$POCL_t^K$	0.8297	0.0049
5	$DOCL_t^K$	0.8368	0.0071
6	$DOCL_t^F$	0.8358	0.0010

Il processo RFS è concluso: per spiegare la dinamica di g_t^{sed}, sono state selezionate complessivamente una variabile di stato (TSS_t^{-3}) e sette forzanti esogene ($SSOL1_t$, $SSOL2_t$, $SSOL3_t$, $SSOL4_t$, $SSOL5_t$, $POCL_t$ e $DOCL_t$).

4.4.4 Deficit irriguo g_t^{irr}

L'ultimo caso da considerare riguarda l'applicazione dell'algoritmo IFR nel caso dell'output g_t^{irr}. I risultati sono mostrati in Tabella 4.16. Ciò che si nota ad una prima analisi è un comportamento molto simile a quello relativo a g_t^{temp}, in particolare per quanto riguarda l'andamento decrescente non monotono di ΔR^2, la selezione di features che poco aggiungono alla spiegazione della dinamica dell'output, e la comparsa di punteggi SISO negativi a partire dalle prime iterazioni dell'algoritmo IFR (vedi Appendice C, Tabella C.4). Per quanto riguarda le variabili di stato selezionate, si nota la presenza del livello h_t, anche se solamente alla dodicesima iterazione (ormai prossimi all'over-fitting) e con un contributo piuttosto esiguo in termini di aumento delle prestazioni del modello ($\Delta R^2 = 4\%$). Compaiono anche gli stati $hTaff_t$ e $gateTaff_t$, in corrispondenza delle ultime due iterazioni e con un ΔR^2 piuttosto basso. Le dinamiche di queste tre variabili di stato sono già state spiegate in precedenza (vedi par. 4.4.1, Tabelle 4.12, 4.8 e 4.7), per cui il modello di g_t^{irr} è completo. Si nota anche la comparsa dei controlli u_t^{-3} e u_t^{-13}, chiaramente correlati al deficit irriguo. In particolare, il controllo u_t^{-3} aumenta fortemente le prestazioni del modello ($\Delta R^2 = 33\%$, contro il 3.6% di u_t^{-13}).
Per quanto riguarda le forzanti esogene selezionate, invece, il comportamento dell'algoritmo IFR lascia alcuni dubbi. Oltre all'assenza dei volumi in ingresso dagli affluenti, si nota la presenza di numerose variabili scarsamente correlate al deficit, come ad esempio le temperature degli afflussi e le loro concentrazioni di ossigeno disciolto e di NH_4, il cui contributo al miglioramento delle prestazioni del modello è essenzialmente scarso. Sono presenti anche alcune informazioni meteo, come $Air-Temp_t$, $Vap-Press_t$ e SW_t, che possono fungere da proxy dell'evaporazione dal serbatoio: tuttavia, il loro contributo complessivo non supera il 4%.

Tabella 4.16: Feature selezionate e prestazioni corrispondenti dei modelli MISO ottenuti tramite l'algoritmo IFR nel caso di g_t^{irr}. Le variabili di stato sono segnate in grassetto.

Iterazione	Feature selezionata	Prestazione MISO (R^2)	ΔR^2
1	T_t^F	0.1392	-
2	$\mathbf{u}_t^{-13}$	0.1752	0.0360
3	$Air-Temp_t$	0.2050	0.0298
4	DO_{t+1}^F	0.2207	0.0157
5	$Vap-Press_t$	0.2478	0.0271
6	$t_{mod\,T}$	0.3321	0.0843
7	T_t^K	0.3337	0.0016
8	$\mathbf{u}_t^{-3}$	0.6396	0.3059
9	$NO_{3,t}^K$	0.7791	0.1395
10	SW_t	0.7809	0.0018
11	DO_t^K	0.7813	0.0004
12	$\mathbf{h}_t$	0.8234	0.0421
13	$NH_{4,t}^K$	0.8301	0.0067
14	$NH_{4,t}^F$	0.8311	0.0010
15	$PO_{4,t}^F$	0.8313	0.0002
16	$Rain_t$	0.8365	0.0052
17	$\mathbf{gateTaff}_t$	0.8417	0.0052
18	$\mathbf{hTaff}_t$	0.8423	0.0006
19	$Cloud-Cover_t$	0.8409	0.0014

Il processo RFS è: per spiegare la dinamica di g_t^{irr}, sono state selezionate complessivamente sei variabili di stato ($hTaff_t$, T_t^{-7}, T_t^{-3}, T_t^{-13}, $gateTaff_t$ ed h_t), il tempo $t_{mod\,T}$, quindici forzanti esogene ($NH_{4,t}$, T_t^K, T_t^F, DO_t^K, DO_t^F, $NO_{3,t}$, $PO_{4,t}$, tre classi di $SSOL_t$, SW_t, $Cloud-Cover_t$, $Vap-Press_t$, $Air-Temp_t$ e $Rain_t$) e i due controlli u_t^{-3} e u_t^{-13}.

4.5 Identificazione dell'emulation model

L'ultimo passo della procedura di riduzione del modello PB richiede l'identificazione del modello ridotto. Come già detto nel par. 3.5.2, l'emulation model non verrà usa-

to direttamente all'interno dell'algoritmo di ottimizzazione: allo scopo di risolvere il problema di controllo, verranno utilizzate le indicazioni sugli stati più significativi emerse durante la costruzione dell'emulation model. Anche se il modello ridotto non sarà necessario nella successiva fase di ottimizzazione, il presente paragrafo mostra i risultati ottenuti nella costruzione dell'emulation model, come referenza per ulteriori studi.
Lo scopo della fase di identificazione dell'emulation model è selezionare due classi di modello per la funzione di trasformazione di uscita $\tilde{\mathbf{h}}_t(\cdot)$ e la funzione di transizione di stato $\tilde{\mathbf{f}}_t(\cdot)$ delle eqs. (1.3a) e (1.3b), che devono essere calibrate e validate tramite il dataset $\mathcal{T}$. Nel nostro caso, la funzione di trasformazione di uscita è data dai modelli identificati per ognuno dei costi per passo g_t (vedi par. 4.4), mentre per quanto riguarda la scelta della classe del modello per $\tilde{\mathbf{f}}_t(\cdot)$, si è deciso di adottare gli Extra-Trees, poiché tale classe ha fornito buone prestazioni complessive (in termini di R^2, eq. (4.1)) durante la fase di riduzione. Per quanto riguarda i parametri, il numero M di alberi e la cardinalità minima n_{min} sono stati lasciati pari a 500 e 50 rispettivamente, mentre K, il numero di cut-directions alternative valutate durante la suddivisione di un nodo, è stata posta uguale al numero di input che caratterizzano ogni modello. I modelli sono calibrati tramite previsione ad un passo, e valutati tramite k-fold cross-validation, con un valore di k pari a 2 (a differenza della fase precedente, dove $k = 10$). Infine, diversamente dalla fase di riduzione, non è stato adottato alcun tipo di campionamento dei dati.

4.5.1 Temperatura in uscita g_t^{temp}

La Tabella 4.17 riporta gli output dell'emulation model di g_t^{temp} e le prestazioni corrispondenti, in termini di R^2, Root Mean Square Error (RMSE) e Percent Error in Peak (PEP).
Il coefficiente di terminazione R^2 è già stato definito nel paragrafo 4.4. L'RMSE è un coefficiente non negativo e non limitato superiormente, che rappresenta il livello complessivo di *agreement* tra i dati osservati Q_i e quelli misurati $\hat{Q}_i$:

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(Q_i - \hat{Q}_i)^2}{n}} \tag{4.2}$$

dove n è il numero dei dati disponibili.

Il PEP esprime, in percentuale, la differenza tra il più alto valore all'interno del dataset modellizzato e il più alto valore nel dataset osservato, normalizzata rispetto al valore di picco del dataset osservato:

$$PEP = \frac{max(Q_i) - max(\hat{Q})_i}{max(Q_i)} \cdot 100 \tag{4.3}$$

Dalla tabella 4.17, si osserva che le prestazioni del modello ridotto sono migliori rispetto alla fase precedente (vedi par. 4.4.1): questo è dovuto all'utilizzo del contenuto informativo dei dati completi, che supplisce al numero più basso di *folds* utilizzato in questa fase. La figura 4.2 - Pannello (a) mette a confronto le traiettorie dell'emulation model e di DYRESM-CAEDYM per la variabile g_t^{temp}. Si nota un buon livello di approssimazione dell'emulation model ai dati di partenza, ma anche la tendenza alla sottostima sistematica dei picchi più alti: ciò è dovuto all'utilizzo degli Extra-Trees, che per loro costruzione tendono mediamente a sottostimare, in quanto considerano il valore medio dell'output all'interno di ogni foglia (per maggiori dettagli sul funzionamento degli Extra-Trees, vedi App. A). In particolare, l'errore di sottostima sul picco tende a diminuire aggiungendo il controllo u_t^{-13} agli output del modello MISO (4.2 - Pannelli (b)-(c)).
La tabella 4.17 riporta anche i valori per le altre variabili di stato del modello di g_t^{temp}.

Tabella 4.17: Struttura e prestazioni (R^2, RMSE e PEP in k-fold cross-validation) degli otto modelli MISO che compongono l'emulation model di g_t^{temp}.

Variabile di output	Variabili di input	R^2	RMSE	PEP
g_t^{temp}	Tab. 4.5	0.7400 (-)	6.9566 (°C^2)	-39.5716 (-)
T_{t+1}^{sed}	Tab. 4.6	0.9270 (-)	1.2354 (°C)	-30.1762 (-)
$gateTaff_{t+1}$	Tab. 4.7	0.6840 (-)	5.7935 (-)	-15.7423 (-)
$hTaff_{t+1}$	Tab. 4.8	0.8438 (-)	4.0473 (m)	-40.3767 (-)
T_{t+1}^{-7}	Tab. 4.9	0.9244 (-)	1.5829 (°C)	-20.1513 (-)
T_{t+1}^{-3}	Tab. 4.10	0.9340 (-)	1.5604(°C)	-14.9793 (-)
T_{t+1}^{-13}	Tab. 4.11	0.9096 (-)	1.6817 (°C)	-22.0586 (-)
h_{t+1}	Tab. 4.12	0.9993 (-)	0.2143 (m)	-0.1632 (-)

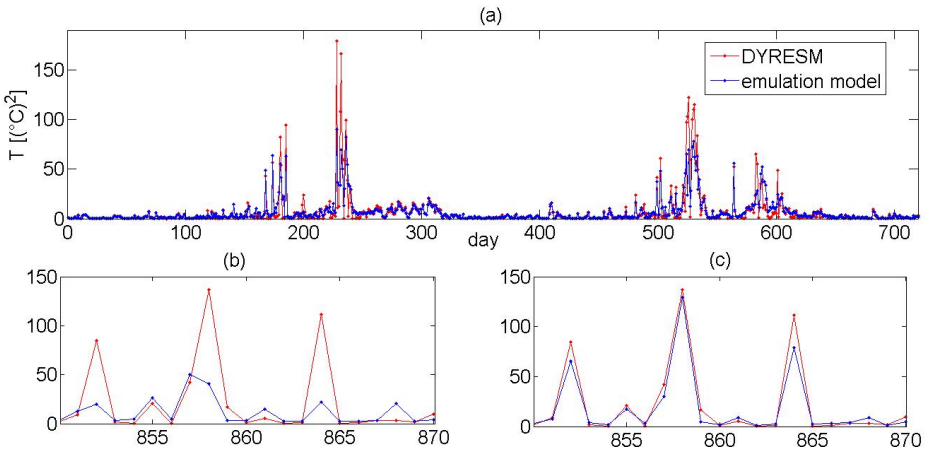

Figura 4.2: Traiettorie di g_t^{temp} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1996) (pannello (a)); comportamento sul picco prima (pannello (b)) e dopo (pannello (c)) l'aggiunta di u_t^{-13} all'emulation model.

4.5.2 Clorofilla nel serbatoio g_t^{rec}

La Tabella 4.18 riporta le prestazioni in k-fold cross-validation del modello ridotto di g_t^{rec} e i valori relativi alle variabili di stato selezionate durante la fase di riduzione, mentre la figura 4.3 mette a confronto le traiettorie dell'emulation model e di DYRESM-CAEDYM.

Tabella 4.18: Struttura e prestazioni (R^2, RMSE e PEP in k-fold cross-validation) dei due modelli MISO che compongono l'emulation model di g_t^{rec}.

Variabile di output	Variabili di input	R^2	RMSE	PEP
g_t^{rec}	Tab. 4.13	0.9455 (-)	0.9535 (g/m^3)	-39.5716 (-)
h_{t+1}	Tab. 4.12	0.9993 (-)	0.2143 (m)	-0.1632 (-)

4.5.3 Sedimentazione nel serbatoio g_t^{sed}

La Tabella 4.19 riporta le prestazioni in k-fold cross-validation del modello ridotto di g_t^{sed} e i valori relativi alle variabili di stato selezionate durante la fase di riduzione, mentre la figura 4.4 mette a confronto le traiettorie dell'emulation model e di DYRESM-CAEDYM.

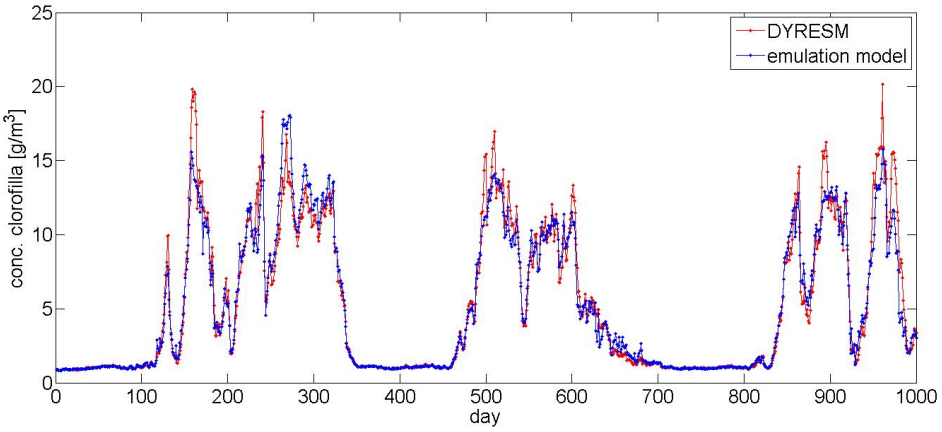

Figura 4.3: Traiettorie di g_t^{rec} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

Tabella 4.19: Struttura e prestazioni (R^2, RMSE e PEP in k-fold cross-validation) dei due modelli MISO che compongono l'emulation model di g_t^{sed}.

Variabile di output	Variabili di input	R^2	RMSE	PEP
g_t^{sed}	Tab. 4.14	0.9930 (-)	$2.0619 \cdot 10^7$ (g/day)	78.3038 (-)
TSS_{t+1}^{-3}	Tab. 4.15	0.8779 (-)	17.1393 (g/m^3)	-54.4243 (-)

4.5.4 Deficit irriguo g_t^{irr}

La Tabella 4.20 riporta le prestazioni i k-fold cross-validation dell'emulation model di g_t^{irr} e i valori relativi alle variabili di stato selezionate durante la fase di riduzione, mentre la figura 4.5 mette a confronto le traiettorie del modello ridotto e di DYRESM-CAEDYM.

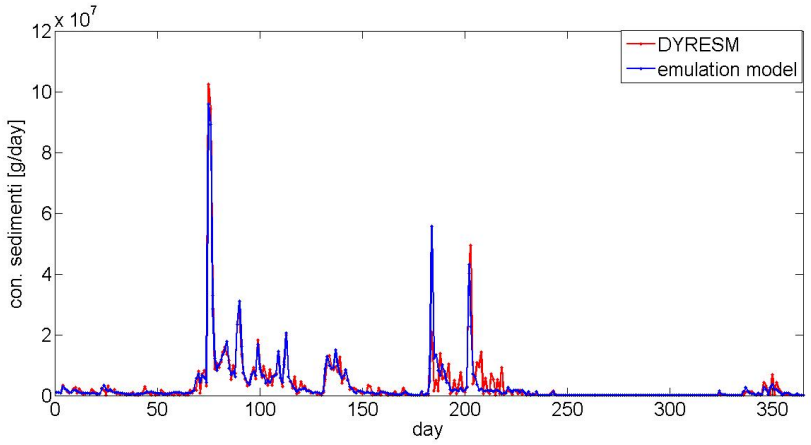

Figura 4.4: Traiettorie di g_t^{sed} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (anno 1995).

Tabella 4.20: Struttura e prestazioni (R^2, RMSE e PEP in k-fold cross-validation) dei sette modelli MISO che compongono l'emulation model di g_t^{irr}.

Variabile di output	Variabili di input	R^2	RMSE	PEP
g_t^{irr}	Tab. 4.16	0.9279 (-)	$5.4632 \cdot 10^7$ $((m^3/s))^2$	-54.3402 (-)
$hTaff_{t+1}$	Tab. 4.8	0.8438 (-)	4.0473 (m)	-40.3767 (-)
T_{t+1}^{-7}	Tab. 4.9	0.9244 (-)	1.5829 (°C)	-20.1513 (-)
T_{t+1}^{-3}	Tab. 4.10	0.9340 (-)	1.5604(°C)	-14.9793 (-)
T_{t+1}^{-13}	Tab. 4.11	0.9096 (-)	1.6817 (°C)	-22.0586 (-)
h_{t+1}	Tab. 4.12	0.9993 (-)	0.2143 (m)	-0.1632 (-)
$gateTaff_{t+1}$	Tab. 4.7	0.6840 (-)	5.7935 (-)	-15.7423 (-)

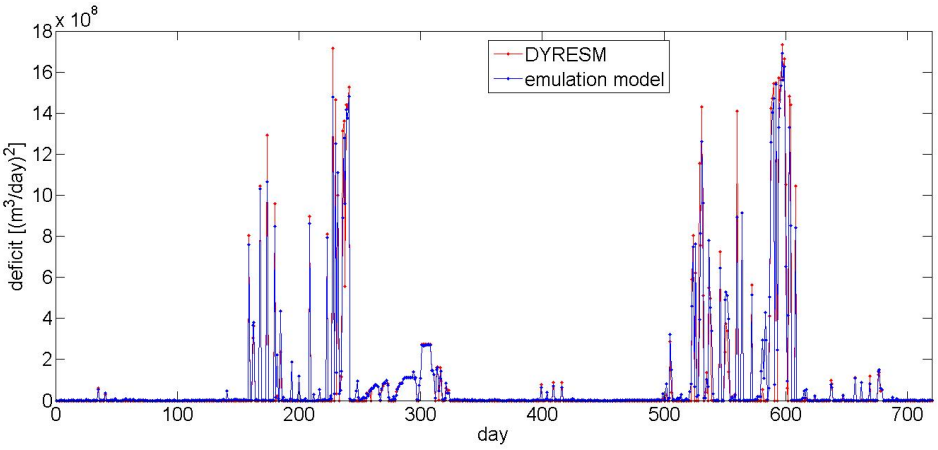

Figura 4.5: Traiettorie di g_t^{irr} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1996).

Capitolo 5

Ottimizzazione

Una volta concluso il processo di costruzione dell'emulation model, è possibile risolvere il problema di controllo ottimo utilizzando due approcci differenti (vedi par. 3.5.2):

- Utilizzare direttamente l'emulation model per la risoluzione del problema di controllo: si usa cioè l'emulation model identificato al passo precedente (che comprende solamente $\tilde{n}_x$ variabili di stato) che si suppone sia in grado di sostituire completamente il modello DYRESM-CAEDYM all'interno dell'algoritmo di ottimizzazione;
- Utilizzare solamente le indicazioni sugli stati più significativi emerse durante la costruzione dell'emulation model per risolvere il problema di controllo, usando i dati originariamente generati dal modello DYRESM-CAEDYM per la costruzione del dataset $\mathcal{F}$.

Nell'ambito di questo lavoro di tesi, si è scelto di adottare la seconda modalità operativa. Le informazioni relative agli stati più significativi e ai controlli verranno estrapolate dalle indicazioni fornite dall'identificazione dell'emulation model (par. 4.4 e 4.5), e gli argomenti del dataset $\mathcal{F}$ in input all'algoritmo di fitted Q-iteration verranno selezionati dai dati generati dal modello DYRESM-CAEDYM (par. 4.1). I risultati della fase di ottimizzazione verranno infine confrontati con quelli ottenuti riducendo lo stato tramite l'utilizzo di un approccio empirico *expert-based* (Castelletti *et al.*, 2010b).

5.1 Definizione degli esperimenti

Allo scopo di generare il dataset di apprendimento $\mathcal{F}$ definito dall'eq. (3.13) e richiesto dall'algoritmo di fitted Q-iteration per il progetto di politiche di controllo mono-obiettivo, è necessario definire il vettore di stato ridotto $\tilde{\mathbf{x}}_t$ che costituisce l'argomento della legge di controllo (3.12).

5.1.1 Vettore di stato ridotto

Come anticipato, le politiche di controllo verranno valutate in funzione del vettore di stato ridotto identificato a partire dalla costruzione dell'emulation model, e i risultati verranno confrontati con quelli delle politiche ottenute in funzione del vettore di stato ridotto identificato tramite l'adozione di un approccio *expert-based*.

Vettore di stato *expert-based*

La riduzione dello stato effettuata adottando l'approccio *expert-based* individua, tra le variabili di stato generate in origine dal modello DYRESM-CAEDYM, quelle che appaiono più significative nel condizionare le decisioni di rilascio, e di conseguenza le politiche di controllo. Sulla base di lavori precedenti (Castelletti *et al.*, 2010b; Galli, 2010; Garbarini, 2009), è stata identificata una formulazione del vettore di stato ridotto $\tilde{\mathbf{x}}_t$ che include il tempo t, il livello h_t del serbatoio, e le temperature e i Solidi Sospesi Totali degli strati corrispondenti ai sifoni direttamente controllati (nell'ambito del presente lavoro, i sifoni a -3 e a -13 m di profondità), per un totale di sei variabili di stato:

$$\tilde{\mathbf{x}}_t^1 = [t, h_t, T_t^{-3}, T_t^{-13}, TSS_t^{-3}, TSS_t^{-13}] \tag{5.1}$$

Vettore di stato derivato da *emulation modelling*

Restano ora da definire le formulazioni di $\tilde{\mathbf{x}}_t$ identificate a partire dalle indicazioni fornite dall'emulation model. In primo luogo, i risultati dell'emulation model relativo a g_t^{sed} evidenziano come la dinamica di questo costo per passo non dipenda dal controllo (par. 4.4.3): per questo motivo, l'indicatore $J^{sed}(p)$ non verrà considerato in questa fase, dal momento che è insensibile al controllo. Ne consegue l'esclusione delle concentrazioni di solidi sospesi dalle variabili dello stato ridotto, in particolare

della variabile TSS_t^{-3} che compare nella descrizione della dinamica di g_t^{sed}.
Per quanto riguarda g_t^{rec} e g_t^{irr}, i rispettivi emulation model suggeriscono unicamente l'adozione del livello h_t. In particolare, h_t risulta essere l'unica variabile di stato selezionata per g_t^{rec} durante la fase di riduzione (par. 4.4.2), mentre nel caso di g_t^{irr} rappresenta la variabile di stato che aumenta maggiormente le prestazioni del modello MISO (par 4.4.4, Tabella 4.16). Il vettore di stato ridotto utilizzato in corrispondenza delle politiche relative a questi costi per passo è il seguente:

$$\tilde{\mathbf{x}}_t^2 = [t, h_t] \tag{5.2}$$

Per quanto riguarda g_t^{temp}, i risultati dell'emulation model (par. 4.4.1) suggeriscono di adottare il profilo di temperatura all'interno del serbatoio, per un vettore $\tilde{\mathbf{x}}_t$ definito come:

$$\tilde{\mathbf{x}}_t^3 = [t, h_t, T_t^{-3}, T_t^{-7}, T_t^{-13}, T_t^{sed}] \tag{5.3}$$

Infine, i controlli $\mathbf{u}_t$ che andranno a completare il dataset $\mathcal{F}$ (eq. (3.13)) sono quelli identificati durante la fase di *Design of Experiments* (par. 4.2), ovvero le decisioni di rilascio dai sifoni a -3 e a -13 m di profondità.

5.1.2 Generazione delle tuple

Una volta definiti i vettori di stato ridotto $\tilde{\mathbf{x}}_t^i$ (i=1,2,3) e il vettore delle decisioni di rilascio $\mathbf{u}_t$ (le cui componenti sono u_t^{-3} e u_t^{-13}), è possibile procedere alla costruzione del dataset $\mathcal{F}$. Oltre alle informazioni su stato ridotto e controllo, le tuple di $\mathcal{F}$ comprendono il vettore di stato $\tilde{\mathbf{x}}_{t+1}$ prodotto dalle transizioni tra t e $t+1$, partendo da $\tilde{\mathbf{x}}_t$ e applicando la decisione $\mathbf{u}_t$, e i costi per passo associati g_{t+1} definiti nel par. 3.2. Le tuple sono state generate tramite il modello DYRESM-CAEDYM (vedi par. 4.2) sul periodo 1995-2006 per 100 differenti scenari della decisione di rilascio $\mathbf{u}_t$, generati in modo pseudo-random con lo scopo di esplorare in maniera più omogenea possibile lo spazio di stato. I costi per passo sono stati calcolati in corrispondenza di tutte le transizioni di stato, e aggregati con pesi differenti in funzione della politica di controllo da generare (vedi par. 5.1.3).

5.1.3 Definizione delle politiche

La presenza di differenti indicatori, corrispondenti a differenti interessi settoriali (vedi par. 3.2), può essere formalizzata definendo una funzione $g_t = g_t(\tilde{\mathbf{x}}_t, \mathbf{u}_t)$ associata alla transizione di stato tra t e $t+1$. Tale funzione può essere ottenuta tramite una somma pesata (Metodo dei Pesi) dei tre costi per passo g_t^i (con $i = 1, \ldots, 3$) considerati:

$$g_t(\tilde{\mathbf{x}}, \mathbf{u_t}) = \sum_{i=1}^{3} \lambda^i g_t^i(\tilde{\mathbf{x}}, \mathbf{u_t}) \tag{5.4}$$

dove $\sum_{i=1}^{3} \lambda^i = 1$ con $\lambda^i \geq 0 \forall i$.
Le politiche considerate nel presente lavoro corrispondono ai punti estremi della frontiera di Pareto, ovvero le politiche ottenute settando a zero tutti i pesi dell'eq. (5.4), tranne quello corrispondente al settore (tra i tre considerati) la cui soddisfazione deve essere massimizzata dalla politica che si sta progettando. Più precisamente, le politiche considerate nel presente lavoro sono le seguenti:

- politica p^{rec} (*Ricreazione*), ottenuta con i seguenti pesi: $|1\ 0\ 0|$ (ovvero i seguenti costi per passo aggregati: $g_t = 1 \cdot g_t^{rec} + 0 \cdot g_t^{irr} + 0 \cdot g_t^{temp}$);
- politica p^{irr} (*Irrigazione*), ottenuta con i seguenti pesi: $|0\ 1\ 0|$;
- politica p^{temp} (*Temperatura*), ottenuta con i seguenti pesi: $|0\ 0\ 1|$.

calcolate in funzione del vettore di stato ridotto $\tilde{\mathbf{x}}_t^1$ (eq. (5.1)), e le politiche

- politica p_{em}^{rec} (*Ricreazione*), pesi: $|1\ 0\ 0|$;
- politica p_{em}^{irr} (*Irrigazione*), pesi: $|0\ 1\ 0|$;
- politica p_{em}^{temp} (*Temperatura*), pesi: $|0\ 0\ 1|$.

calcolate in funzione dei vettori di stato ridotti $\tilde{\mathbf{x}}_t^2$ (eq. (5.2), politiche p_{em}^{rec} e p_{em}^{irr}) e $\tilde{\mathbf{x}}_t^3$ (eq. (5.3), politica p_{em}^{temp}).

5.2 Risultati

I valori degli indicatori ottenuti simulando sul periodo di validazione (1990-1994) le tre politiche post-emulation model p^i_{em} corrispondenti agli estremi della frontiera di Pareto sono riportati in Tabella 5.1, in comparazione con i valori delle politiche p^i relative allo stato ridotto identificato dall'approccio *expert-based*. Per il calcolo delle politiche, è stato utilizzato l'algoritmo di fitted Q-iteration (Ernst *et al.*, 2005, che si basa su una regressione tree-based (App. B). I valori degli indicatori associati sono stati calcolati tramite simulazione del modello DYRESM-CAEDYM.

Tabella 5.1: Riassunto delle performances delle politiche *expert-based* e post-em, espresse in termini di valori degli indicatori. Gli indicatori sono espressi come la media sull'intero periodo di validazione (1990-1994).

Settore	Indicatore	U.M.	Pol. *expert-based*	Politica post-em
Ricreazione	J^{rec}	g/m^3	2.1469E+00	4.8555E+00
Irrigazione	J^{irr}	$(m^3/day)^2$	2.2153E+07	2.2819E+07
Temperatura	J^{temp}	$(°C)^2$	3.0850E+00	1.8166E+00

5.2.1 Temperatura

Il miglioramento ottenuto dalla politica post-em p^{temp}_{em} rispetto alla politica *expert-based* p^{temp} (1.8166 vs 3.0850) risulta essere piuttosto significativo. L'utilizzo del vettore di stato ridotto $\tilde{\mathbf{x}}^3_t$ (eq. (5.3)), che considera il profilo di temperatura all'interno del serbatoio, sembra possedere un contenuto informativo decisamente migliore di $\tilde{\mathbf{x}}^1_t$ (eq. (5.1)) relativamente al calcolo del valore di J^{temp}: se si considera unicamente il settore *Temperatura*, l'informazione relativa alle temperature degli strati a -7 m di profondità e a 18 m dal fondo (T^{-7}_t e T^{sed}_t) migliora fortemente le prestazioni della politica. In particolare, dalla fig. 5.1 - Pannello (a), si nota come le differenze di temperatura tra afflusso e rilascio siano mediamente più contenute nel caso di p^{temp}_{em}. In fig. 5.1 - Pannello (b) sono messe a confronto le temperature dell'afflusso e le temperature del rilascio calcolate tramite le due politiche: si può notare come la temperatura in uscita relativa a p^{temp}_{em} si avvicini maggiormente alla temperatura dell'afflusso, rispetto alla temperatura in uscita relativa a p^{temp}.

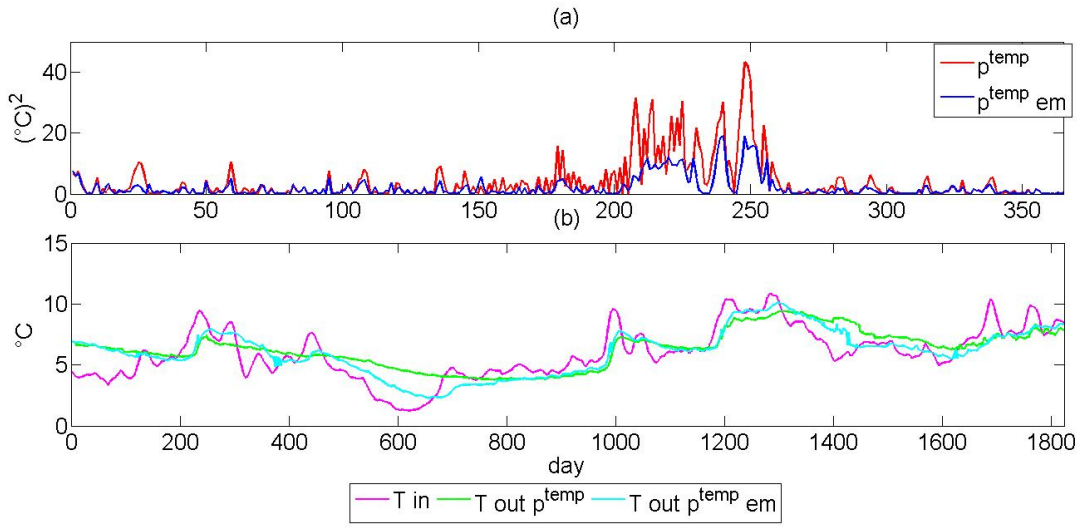

Figura 5.1: Traiettorie del costo per passo g_t^{temp} del settore *Temperatura* (differenza quadratica giornaliera tra le temperature dell'afflusso e del rilascio) ottenute simulando le politiche p^{temp} e p_{em}^{temp} (anno 1990) (pannello (a)); temperatura dell'afflusso, e temperature del rilascio corrispondenti a p^{temp} e p_{em}^{temp} (periodo 1990-1994) (pannello (b)).

5.2.2 Irrigazione

La comparazione tra le politiche p^{irr} e p_{em}^{irr} relative al settore *Irrigazione* mostra all'incirca lo stesso risultato in termini di prestazione della politica (vedi Tabella 5.1). Il valore di J^{irr}, espresso in $(\mathrm{m}^3/\mathrm{day})^2$, risulta essere dell'ordine di 10^7 per entrambe le politiche. Per avere un'idea delle effettive prestazioni delle politiche, è utile esprimere l'indicatore in m^3/s: si ottengono così i valori $5.4475 \cdot 10^{-2}$ e $5.5288 \cdot 10^{-2}$, rispettivamente per p^{irr} e p_{em}^{irr}. Ciò dimostra quanto i valori di deficit siano in realtà molto piccoli, quasi trascurabili. La figura 5.2 mostra le traiettorie di g_t^{irr} generate dalle due politiche: si nota come si abbia realmente un deficit solamente in due periodi ben definiti dell'orizzonte di validazione (periodi estivi 1990 e 1994). Allo stesso modo, in figura 5.3 - Pannello (a) si nota come il rilascio dal serbatoio di Tono sia quasi sempre ampiamente superiore alla domanda lungo tutto l'orizzonte di validazione, fatta eccezione nelle estati del 1990 e del 1994, periodi in cui si produce effettivamente un deficit (fig. 5.3 - Pannelli (b) e (c)). Il fatto che la politica post-emulation model p_{em}^{irr} abbia le stesse prestazioni della politica *expert-based* p^{irr} suggerisce che, per quanto riguarda il settore *Irrigazione*, la politica *expert-based* utilizzi quattro informazioni di stato superflue.

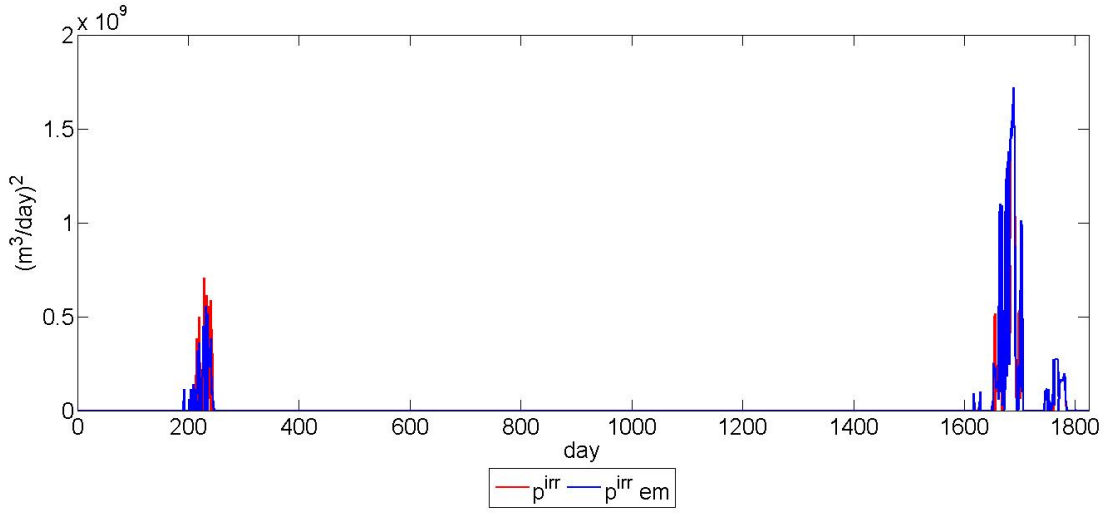

Figura 5.2: Traiettorie del costo per passo g_t^{irr} del settore *Irrigazione* (deficit idrico quadratico giornaliero) ottenute simulando le politiche p^{irr} e p_{em}^{irr} sull'intero orizzonte di validazione (periodo 1990-1994).

5.2.3 Ricreazione

Le prestazioni della politica p_{em}^{rec}, valutata in funzione del vettore di stato ridotto $\tilde{\mathbf{x}}_t^2$ (eq. (5.2)), mostra un netto peggioramento rispetto alla politica p^{rec}, valutata in funzione dello stato ridotto $\tilde{\mathbf{x}}_t^1$ identificato tramite l'approccio *expert-based* (eq. (5.1)). In particolare, la figura 5.4 mostra come la traiettoria di g_t^{rec} ottenuta dalla politica p_{em}^{rec} differisca fortemente da quella calcolata tramite la politica p^{rec}: in corrispondenza dei picchi di crescita algale, la prima mostra concentrazioni medie giornaliere più alte (circa raddoppiate) rispetto alla seconda. La spiegazione più plausibile del fenomeno è che alcune delle variabili di stato contenute in $\tilde{\mathbf{x}}_t^1$ forniscano un'informazione supplementare utile per la corretta valutazione della politica relativa al settore *Ricreazione*: in mancanza di questa informazione, le prestazioni della politica p_{em}^{rec} risultano essere nettamente peggiori. In particolare, la temperatura T_t^{-3} relativa al *layer* a -3 m di profondità, può essere considerata un proxy dell'irraggiamento solare: infatti, la luce è un fattore limitante della crescita algale (par. 4.4.2), e la concentrazione di Chl-a viene misurata nello strato superficiale, dove le fioriture algali sono favorite (par. 3.3.1) rispetto agli strati profondi in cui la luce penetra con più difficoltà.
Inoltre, il modello di g_t^{rec} (par. 4.4.2) dipende fortemente da variabili esogene (DO, NH_4, irraggiamento solare) che non fanno parte degli argomenti della politica: è

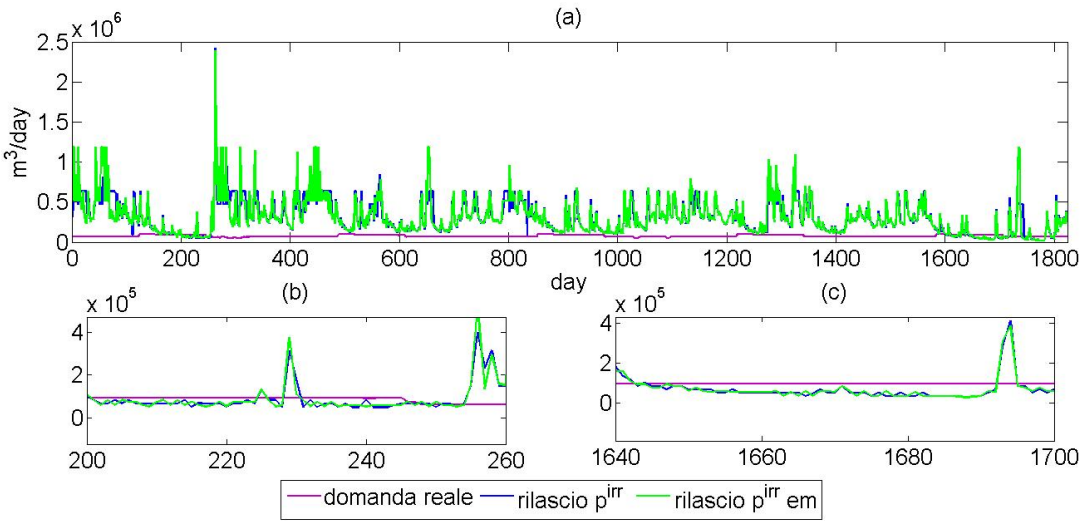

Figura 5.3: Traiettorie della domanda irrigua e dei rilasci calcolati tramite le politiche p^{irr} e p^{irr}_{em} (periodo 1990-1994) (Pannello (a)); deficit estate 1990 (Pannello (b)); deficit estate 1994 (Pannello (c))

probabile dunque che le prestazioni della politica risentano dell'assenza di queste variabili.

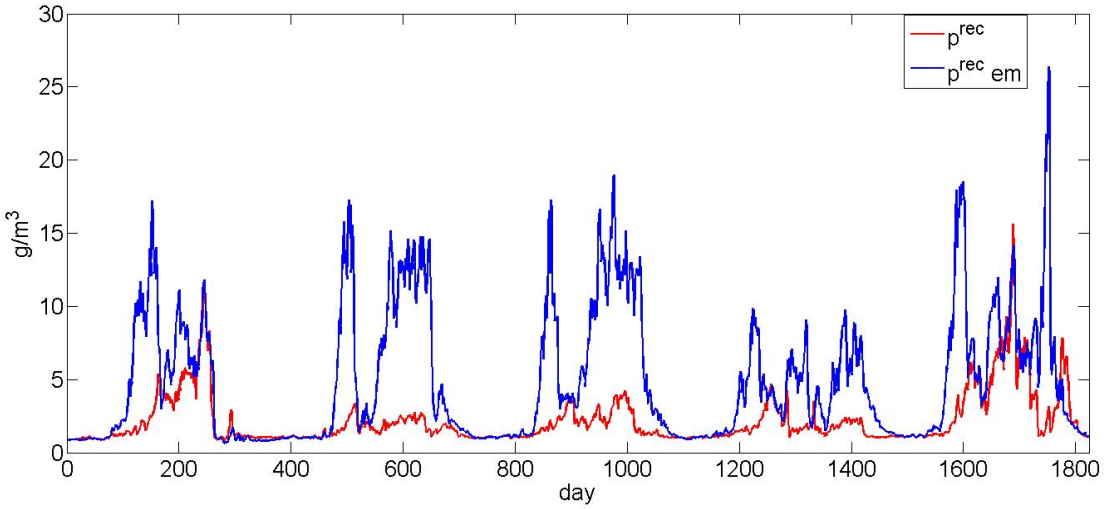

Figura 5.4: Traiettorie del costo per passo g_t^{rec} del settore *Ricreazione* (concentrazione massima giornaliera di chl-a) ottenute simulando le politiche p^{rec} e p_{em}^{rec} sull'intero orizzonte di validazione (periodo 1990-1994).

Conclusioni

Gli effetti prodotti dai controlli delle politiche di regolazione progettate hanno dimostrato la percorribilità dell'approccio proposto. Si può vedere come le prestazioni relative alla politica per il settore *Temperatura* calcolate in base all'utilizzo delle informazioni fornite dall'emulation model siano migliori rispetto a quelle calcolate utilizzando l'approccio *expert-based*: per questo settore, le indicazioni sul profilo di temperatura all'interno del serbatoio, emerse durante l'identificazione dell'emulation model, sono più significative di quelle relative ai solidi sospesi totali fornite dall'approccio *expert-based*. Di contro, si osserva un certo peggioramento per quanto riguarda la politica relativa al settore *Ricreazione*, per la quale l'approccio *expert-based* fornisce risultati migliori: questo è dovuto al mancato utilizzo delle informazioni sulla temperatura da parte della politica post-emulation model, ma anche al fatto che quest'ultima non considera le informazioni sulle forzanti esogene suggerite dalla costruzione dell'emulation model. Infine, si nota come le prestazioni della politica relativa al settore *Irrigazione* restino praticamente invariate utilizzando entrambi gli approcci: si può perciò ritenere che le informazioni su temperatura e solidi sospesi suggerite dall'approccio *expert-based* siano ridondanti. Va detto anche che i valori di deficit calcolati sul periodo di validazione 1990-1994 sono per lo più circoscritti a due periodi specifici (periodi estivi 1990 e 1994), e comunque poco significativi, viste le esigue portate in gioco. Per questo motivo, un futuro sviluppo di questo studio potrebbe riguardare il confronto tra politiche post-emulation model ed *expert-based* nel caso di deficit più significativi, e quindi con domanda idrica più elevata.
Un altro possibile sviluppo del presente lavoro può essere quello di integrare gli obiettivi di quantità e qualità dell'acqua, calcolando alcuni punti appartenenti alla frontiera di Pareto, senza limitarsi agli estremi. I risultati ottenuti potrebbero fornire ulteriori prove della bontà dell'approccio utilizzato nel presente lavoro, incentivando futuri sviluppi della metodologia adottata. In particolare, un aspetto che rimane inesplorato in conclusione di questo lavoro di tesi è quello relativo alla possibilità di

utilizzare la struttura dell'emulation model direttamente all'interno del framework di ottimizzazione, senza limitarsi all'uso delle indicazioni fornite dallo stesso modello ridotto sulle variabili di stato più significative nello spiegare la dinamica dei costi per passo. Inoltre, un ulteriore sviluppo potrebbe consistere nel considerare, tra gli argomenti delle politiche, anche le informazioni esogene: questo sarebbe molto utile nel caso della clorofilla, il cui modello dipende fortemente da alcune forzanti esogene.

Appendice A

Extra-Trees e feature ranking basato sugli extra-trees

A.1 Panoramica dei metodi tree-based

I metodi tree-based rappresentano una classe di metodi di regressione (e classificazione) non parametrici in grado di fornire flessibilità di modellizzazione, efficienza computazionale, interpretabilità e un buon grado di accuratezza. Sono tutti basati sul concetto di albero decisionale, ovvero una struttura rappresentante una cascata di regole che portano a valori numerici. Queste strutture, composte da nodi decisionali, rami e foglie, sono ottenuti partizionando, in corrispondenza del nodo decisionale più alto e tramite un criterio di separazione appropriato, l'insieme dei regressori in due subset, creando quindi i primi due rami. Il processo di divisione viene poi ripetuto in maniera ricorsiva in corrispondenza di ogni subset derivato, fino a che il valore numerico associato ad un subset varia solo leggermente, oppure finché rimangono pochi elementi. Quando questo processo termina, i rami rappresentano la struttura gerarchica della divisione dei subset, mentre le foglie sono i subset più precisi associate ai rami terminali. Ad ogni foglia viene infine associato un valore numerico.
Il più popolare metodo basato sugli alberi è il cosiddetto Classification and Regression Trees (CART; Breiman *et al.*, 1984). Il processo costruttivo del CART lavora con un approccio top-down che parte dal nodo decisionale più alto ed esplora sistematicamente l'insieme dei regressori, alla ricerca di nodi da separare. La separazione di un nodo richiede di cercare la migliore combinazione di regressori e valori di separazione che minimizza un certo criterio di separazione, definito in base alle capacità

della regressione attuata dal CART. Il processo termina quando tutti i subset sono dichiarati terminali, cioè quando possiedono valori identici per tutti i regressori o per l'output, oppure quando contengono meno campioni rispetto ad un numero predefinito. Per problemi di regressione il valore previsto dell'output associato ad ogni foglia è la media dei valori degli output misurati. I CART sono veloci, concettualmente semplici, non richiedono la calibrazione di alcun parametro e, nel caso di dataset piccoli, forniscono un'interpretazione fisica della loro struttura. Il limite principale dei CART è la natura costante e piece-wise dell'output previsto, a causa dell'attribuzione dello stesso valore dell'output previsto a tutti i campioni contenuti all'interno della stessa foglia, indipendentemente dalla loro posizione nello spazio dei regressori. Inoltre, i modelli indotti dal metodo CART sono caratterizzati da un'elevata varianza dell'output previsto, poiché le separazioni scelte in corrispondenza di ogni nodo interno dipendono in gran misura dalla natura casuale del dataset di addestramento (Wehenkel, 1997; Geurts and Wehenkel, 2000).
Uno sviluppo naturale dei CART si basa sull'idea di sostituire i valori numerici costanti in corrispondenza delle foglie con dei modelli lineari multi-variati, portando quindi alla costruzione di un modello ad albero (M5; Quinlan, 1992), che può essere visto come un modello modulare (Solomatine e Dulal, 2003), con ogni regressione lineare specializzata su di uno specifico subset dell'insieme dei regressori. La differenza principale tra l'approccio dei modelli ad albero e l'uso di combinazioni di modelli specializzati è che il primo, in funzione di un criterio di separazione predefinito, esegue una suddivisione automatica ed ottimale dell'insieme dei regressori. Gli M5, paragonati ai CART, sono più piccoli, più compatti e, inoltre, forniscono una migliore accuratezza previsionale. Quest'ultima proprietà è attribuibile alla capacità del modello di sfruttare la linearità locale nei dati e di estrapolare, ovvero di prevedere quei valori presenti al di fuori del range osservato durante il processo di costruzione.
Un secondo approccio alternativo per migliorare l'accuratezza dei CART è quello di ridurre la loro varianza combinando numerosi alberi, costruendo cioè 'foreste' di alberi. Breiman (1996) ha proposto previsori Bagging, un'istanza particolare di una più ampia famiglia di metodi basati du foreste di alberi, conosciuta nella letteratura del machine learning come *randomization methods*. Questi metodi introducono esplicitamente la randomizzazione nel processo costruttivo e sfruttano, ad ogni sessione, una diversa versione randomizzata del dataset di addestramento originario, producendo quindi una foresta di alberi diversificati. La previsione di questi albe-

ri può essere poi aggregata tramite media. I metodi di randomizzazione possono aumentare considerevolmente l'accuratezza di previsione dei CART e, anche se richiedono di costruire numerosi alberi, rimangono comunque efficienti dal punto di vista computazionale, a causa del basso costo di calcolo del processo standard di costruzione di un albero. Gli sforzi di ricerca si sono concentrati di recente su una randomizzazione diretta del processo costruttivo degli alberi. Questo ha portato allo sviluppo di numerosi metodi (ad esempio, Random subspace method (Ho, 1998); Random forests (Breiman, 2001); PERT (Cutler and Guohua, 2001)) che generano una perturbazione all'interno degli alberi modificando l'algoritmo responsabile della ricerca della suddivisione ottima durante la crescita dell'albero.
Geurts *et al.* (2006) hanno proposto un nuovo metodo basato sulle foreste di alberi, denominato Extremely Randomized Trees (Extra-Trees), che randomizza (totalmente o parzialmente) sia i regressori, sia la selezione dei cut-point quando si suddivide un nodo. È stato empiricamente dimostrato che gli extra-trees superano altri metodi randomizzati (Tree Bagging, Random Subspace Method, Random Forests, PERT) e non randomizzati (CART) in termini di accuratezza di previsione ed efficienza computazionale.

A.2 Gli extra-trees

Il metodo degli Extra-Trees costruisce foreste di alberi di regressione in accordo con il classico approccio top-down. Gli Extra-Trees, diversamente dagli altri metodi di randomizzazione, sfruttano il dataset di addestramento originario, e dividono i nodi selezionando i cut-point e i regressori totalmente (o parzialmente) in maniera random. L'idea dietro queste due caratteristiche è che l'uso del dataset di addestramento originale serva per ridurre il residuo, mentre la randomizzazione dei cut-point e la selezione dei regressori, combinata con la media attuata sulla foresta, può ridurre la varianza in maniera più efficace rispetto ad altri metodi di randomizzazione (vedi Geurts *et al.* (2006)).
La procedura di costruzione degli Extra-Trees, per un problema di regressione con una variabile di output r, n regressori $\{z_1, z_2, \ldots, z_n\}$ e un dataset di addestramento S (composto da N osservazioni) è illustrata in Tabella A.1 per uno degli M alberi che compongono una foresta (per ulteriori dettagli vedi Geurts *et al.*, 2006).
Dal punto di vista computazionale, la complessità della procedura di costruzione della foresta è dell'ordine di $N \log(N)$ rispetto all'ampiezza del dataset di adde-

Tabella A.1: Procedura di costruzione degli Extra-Trees.

Passo 0. Selezionare in maniera random K regressori, $\{z_1, z_2, \ldots, z_K\}$ tra tutti gli n regressori non costanti disponibili in S.

Passo 1. Per ogni regressore z_i selezionato (con $i = 1, \ldots, K$):

- calcolare il valore minimo e massimo di z_i in S, denotati rispettivamente come $z^S_{i,min}$ e $z^S_{i,max}$.
- tracciare un cut-point $z_{i,c}$ in maniera uniforme in $\left[z^S_{i,min}, z^S_{i,max}\right]$.
- restituire la suddivisione $[z_i < z_{i,c}]$.

Passo 2. Tra le K suddivisioni $\{s_1, s_2, \ldots, s_K\}$, selezionare la suddivisione s_* tale che

$$s_* = \arg\max_{i=1,\ldots,K} \text{Score}(s_i, S) \tag{A.1a}$$

dove

$$\text{Score}(s_i, S) = \frac{\text{var}\{r|S\} - \frac{|S_{i,l}|}{|S|}\text{var}\{r|S_{i,l}\} - \frac{|S_{i,r}|}{|S|}\text{var}\{r|S_{i,r}\}}{\text{var}\{r|S\}} \tag{A.1b}$$

dove$S_{i,l}$ e $S_{i,r}$ sono i due subset di S che soddisfano rispettivamente le condizioni $z_i < s_i$ e $z_i \geqslant s_i$, $|S|$ è il numero dei campioni in S, $|S_{i,l}|$ e $|S_{i,r}|$ sono il numero di campioni in $S_{i,l}$ e $S_{i,r}$, $\text{var}\{r|S\}$ è la varianza dell'output r nell'insieme S, $\text{var}\{r|S_{i,l}\}$ e $\text{var}\{r|S_{i,r}\}$ sono le varianze di r nei subset $S_{i,l}$ e $S_{i,r}$.

Passo 3. Suddividere l'insieme S nei subset S_l e S_r, in funzione di s_*.

Passo 4. Per il subset S_l (e S_r) verificare le seguenti condizioni:

- $|S_l|$ (o $|S_r|$) $< n_{min}$, dove n_{min} è il numero di campioni necessario per suddividere un nodo (cardinalità minima).
- tutti i regressori siano costanti in S_l (or S_r).
- l'output sia costante in S_l (o S_r).

Se una di queste condizioni è verificata, il subset è una foglia e non è possibile una sua ulteriore suddivisione; altrimenti i Passi 0-4 devono essere ripetuti sostituendo S con S_l (o S_r). La procedura termina quando tutti i subset sono considerati foglie. Ad ogni foglia viene assegnato un valore, ottenuto tramite media dei valori degli output associati ai regressori che cadono all'interno della foglia stessa. Le stime prodotte dagli M alberi sono infine aggregati tramite media aritmetica.

stramento, mentre il tempo di calcolo aumenta linearmente con M e decresce in maniera logaritmica all'aumentare di n_{min}. Nonostante richieda la costruzione di M alberi, questa procedura rimane comunque efficiente dal punto di vista computazionale, poiché la regola di suddivisione è molto semplice, se paragonata ad altre regole di suddivisione che ottimizzano localmente la scelta dei cut-point. Agli Extra-Trees vengono quindi associati tre parametri, i cui valori possono essere fissati, rispetto alle specifiche del problema, sulla base di valutazioni empiriche o in maniera automatica (ad esempio, tramite cross-validazione):

- K, il numero di *cut directions* alternative valutate ad ogni nodo durante la costruzione dell'extra tree, può essere scelto nell'intervallo $[1, \ldots, n]$, dove n è il numero dei regressori. K influenza la randomizzazione: più piccolo è K, più forte è la randomizzazione degli alberi e più debole la dipendenza della loro struttura dai valori dell'output nel dataset di addestramento. Nel caso estremo, quando K è uguale a n, la scelta della direzione di taglio non è randomizzata, e la randomizzazione agisce solamente attraverso la scelta del cut-point. Al contrario, quando K è uguale a 1, le suddivisioni (*cut directions* e cut-point) vengono scelte in maniera totalmente indipendente dall'output e il metodo costruisce degli alberi totalmente randomizzati. Geurts *et al.* (2006) ha osservato che il residuo decresce in maniera monotona e la varianza aumenta quando K aumenta, perciò, in linea di principio, esiste un trade-off tra le riduzioni di residuo e di varianza. Ciò nonostante, per problemi di regressione, gli stessi autori hanno dimostrato empiricamente che il valore ottimo di per K è n.

- n_{min}, la cardinalità minima necessaria per suddividere un nodo. Grandi valori di n_{min} portano ad alberi piccoli (poche foglie), con elevato residuo e varianza bassa. Di contro, valori bassi di n_{min} portano ad alberi pienamente sviluppati, che potrebbero portare all'over-fitting dei dati. Il valore ottimale di n_{min} dipende non solo dall'avversione al rischio di over-fitting, ma anche dal livello di rumore nell'output del dataset di addestramento: più l'output è rumoroso, più elevato dovrebbe essere il valore ottimale di n_{min}. Geurts *et al.* (2006) ha dimostrato empiricamente che, per i problemi di regressione, un valore di default di n_{min} corrispondente a 5 (anche se leggermente sub-ottimo) è una scelta robusta per un'ampia gamma di condizioni tipiche.

- M, il numero di alberi nella foresta, influenza la forza della riduzione della varianza e il comportamento dell'errore di stima, che è una funzione decrescente di M (Breiman, 2001). L'accuratezza di stima aumenta quindi con M e la scelta del suo valore dipende dal trade-off tra l'accuratezza del modello desiderata e la potenza di calcolo disponibile.

A.3 Feature ranking

La struttura particolare degli Extra-Trees, oltre a fornire buone prestazioni dal punto di vista della riduzione di residuo e varianza, può essere sfruttata per classificare l'importanza degli n regressori nello spiegare il comportamento dell'output e in seguito per identificare le variabili più rilevanti tra le n variabili candidate. Questo approccio, così come proposto da Wehenkel (1998) e Fonteneau *et al.* (2008), si basa sull'idea di assegnare un punteggio ad ogni regressore stimando la riduzione di varianza che può essere ad esso associata propagando il dataset di addestramento S sulle M differenti strutture ad albero che compongono la foresta.
Più precisamente, la rilevanza $G(z_i)$ di ogni regressore z_i può essere valutata con la seguente funzione punteggio:

$$G(z_i) = \frac{\sum_{\tau=1}^{M} \sum_{j=1}^{\Omega} \delta(\nu_j, z_i) \cdot \Delta_{var}(\nu_j)|S|}{\sum_{\tau=1}^{M} \sum_{\nu_j=1}^{\Omega} \Delta_{var}(\nu_j)|S|} \tag{A.2a}$$

dove ν_j è il j-esimo nodo non terminale nell'albero τ, Ω è il numero dei nodi non terminali nell'albero τ, $\delta(\nu_j, z_i)$ è uguale a 1 se z_i è utilizzato per suddividere il nodo ν_j (e uguale a 0 altrimenti), $|S|$ è il numero di campioni all'interno del subset S considerato, $\Delta_{var}(\nu_j)$ è la riduzione di varianza che si ha quando si suddivide il nodo ν_j, ovvero

$$\Delta_{var}(\nu_j) = \mathrm{var}\{r|S\} - \frac{|S_{i,l}|}{|S|}\mathrm{var}\{r|S_{i,l}\} - \frac{|S_{i,r}|}{|S|}\mathrm{var}\{r|S_{i,r}\} \tag{A.2b}$$

dove i termini $S_{i,l}$ and $S_{i,r}$ denotano i due subset di S che soddisfano le condizioni $z_i < s_i$ e $z_i \geqslant s_i$ rispettivamente (vedi eq. (A.1b)). I regressori vengono infine classificati in base ai valori decrescenti della loro rilevanza.

Appendice B

Fitted Q-iteration

B.1 Introduzione al Reinforcement Learning

Nonostante i grandi progressi raggiunti negli ultimi decenni, la gestione ottima dei sistemi idrici rimane un'area di ricerca molto attiva. La combinazione di usi dell'acqua molteplici e conflittuali, le non linearità presenti nei modelli e negli obiettivi, le forti incertezze negli input, e le notevoli dimensioni dello spazio degli stati rendono il problema intrigante e ricco di sfide.
La Programmazione Dinamica Stocastica (SDP) è uno dei metodi più affidabili per progettare politiche di gestione Pareto-ottimali di sistemi di serbatoi (vedi Soncini-Sessa *et al.*, 2007). La SDP è basata sulla formulazione del problema di progetto come un processo decisionale sequenziale. L'idea chiave è quella di usare delle particolari funzioni dette *funzioni valore*, per organizzare e strutturare la ricerca di politiche ottime in domini stocastici. Una decisione presa ora può produrre non solo un costo immediato, ma può anche influenzare il successivo stato del sistema, e attraverso lo stato, tutti i costi successivi. Il concetto base della SDP è perciò quello di guardare ad eventi futuri e calcolare così un valore a priori, che viene poi utilizzato per aggiornare la funzione valore.

Sebbene sia stata studiata in maniera estensiva nella letteratura, la SDP soffre di una duplice problematica che di fatto impedisce la sua applicazione pratica a sistemi idrici ragionevolmente complessi:

1. La complessità di calcolo cresce esponenzialmente con le dimensioni dello stato, dei controlli e dei disturbi (la *Curse of Dimensionality* di Bellman [Bellman,

1957]), di modo che la SDP non possa essere utilizzata per sistemi idrici dove il numero di serbatoi sia superiore a poche unità (2-3 serbatoi);

2. È richiesto un modello esplicito di ogni componente del sistema idrico (*Curse of Modelling* [Bertsekas e Tsitsiklis, 1996]) per anticipare gli effetti delle transizioni del sistema).

Qualsiasi informazione utilizzabile dalla SDP può essere o una variabile di stato descritta da un modello dinamico, oppure un disturbo stocastico, bianco, descritto dalla sua funzione di densità di probabilità. L'informazione esogena $\mathbf{W}_t$ (come ad esempio la temperatura, la precipitazione, l'altezza del manto nevoso), il cui utilizzo potrebbe migliorare notevolmente la gestione del serbatoio (Tejada-Guibert *et al.*, 1995; Hejazi *et al.*, 2008), non può essere considerata esplicitamente, a meno che non si introduca un modello dinamico per ognuna delle informazioni aggiuntive di cui si dispone: in altre parole, trasformando le variabili esogene in variabili di stato, e andando così ad aumentare il contributo della *Curse of Dimensionality*. Inoltre, in grandi reti di serbatoi, i disturbi sono, in generale, correlati nel tempo e nello spazio. Sebbene l'inclusione della variabilità spaziale nel processo di identificazione della pdf dei disturbi sia un'operazione piuttosto complessa in molti casi, questo processo non va ad aumentare la complessità di calcolo. Di contro, la correlazione temporale può essere considerata in maniera appropriata utilizzando un modello dinamico stocastico, che però può contribuire enormemente alla *Curse of Dimensionality*.
In letteratura esistono numerosi tentativi di superare la maledizione della dimensionalità: Dynamic Programming based on Successive Approximation (Bellman e Dreyfus, 1962), Incremental Dynamic Programming (Larson, 1968), Differential Dynamic Programming (Jacobson e Mayne, 1970), e numerose soluzioni euristiche per problemi specifici (Wong e Luenberger, 1968; Luenberger, 1971). Tuttavia, questi metodi sono stati concepiti principalmente per problemi deterministici e sono di scarso interesse per la gestione ottima delle reti di serbatoi reali, dove l'incertezza associata ai processi idro-meteorologici di fondo non può essere trascurata. Gli approcci alternativi possono essere classificati in due gruppi principali (vedi Castelletti *et al.*, 2008 per ulteriori dettagli), a seconda della strategia che adottano per alleviare l'onere della dimensionalità: metodi basati sulla semplificazione del modello del sistema idrico e metodi basati sulla restrizione dei gradi di libertà del problema di progetto della politica.
La prima classe include sia i metodi di scomposizione del sistema completo in sottosistemi più piccoli e trattabili, con il conseguente utilizzo di una procedura iterativa

per la risoluzione del problema, sia l'aggregazione dei sottosistemi, o parte di essi, in un sistema composto e trattabile dal punto di vista computazionale. Per fare un esempio, Turgeon (Turgeon, 1981) ha proposto un algoritmo basato sulla scomposizione di un problema di gestione di N serbatoi in N sottoproblemi, ognuno dei quali considera due serbatoi: uno formulato rispetto al serbatoio in esame, più un secondo problema formulato rispetto ad un serbatoio equivalente che tenga conto di tutti gli invasi a valle. In questo modo, il tempo di calcolo complessivo per la soluzione del problema cresce linearmente con N. L'idea dietro questo approccio è che diversi livelli di scomposizione vengano modellizzati ed analizzati separatamente, ma che qualche tipo di informazione venga trasmessa dai livelli inferiori a quelli superiori della gerarchia di scomposizione.

La seconda classe di approcci per superare la maledizione della dimensionalità è basata sull'introduzione di alcune ipotesi di regolarità della funzione valore ottima della SDP. Siccome la SDP richiede la discretizzazione degli spazi dello stato e delle decisioni ammissibili, un modo per mitigare (ma non per eliminare del tutto) il problema della dimensionalità è combinare una griglia di discretizzazione lasca con un'approssimazione continua della funzione valore. Essendo considerati approssimatori di funzioni universali, le reti neurali artificiali sono particolarmente adatte a questo proposito.

La *Curse of Modelling* della SDP ha ricevuto minore attenzione rispetto alla sua 'gemella'. Nella SDP, i modelli sono richiesti per prevedere e valutare gli effetti di ogni decisione plausibile sulle dinamiche dello stato, calcolandone il costo associato. Un approccio alternativo per effettuare tale valutazione è affidarsi direttamente all'esperienza. Questa è proprio l'idea centrale dell'Apprendimento per Rinforzo o Reinforcement Learning (RL), una procedura molto conosciuta per il decision-making sequenziale (Kaelbling *et al.*, 1996; Barto e Sutton, 1998) che combina concetti dall'SDP, dall'approssimazione stocastica, e dall'approssimazione di funzioni. L'esperienza di apprendimento può essere acquisita in linea, sperimentando direttamente le decisioni sul sistema reale senza l'ausilio di alcun modello, o generata fuori linea, utilizzando un simulatore esterno al processo di ottimizzazione oppure una serie di osservazioni storiche. Mentre la prima opzione è chiaramente impraticabile sulle reti di serbatoi reali, l'apprendimento fuori linea è già stato sperimentato con successo nella gestione dei sistemi idrici. Castelletti *et al.*, 2001 (vedi anche Soncini-Sessa *et al.*, 2007) ha proposto una versione parzialmente model-free del Q-learning classico (Watkins e Dayan, 1992) per il progetto della politica di gestione giornaliera

di un lago regolato. La dinamica dell'invaso è stata simulata tramite un'equazione di bilancio di massa. L'ingresso è stato descritto utilizzando la sequenza storica dell'afflusso. Usando sia l'invaso sia l'afflusso al giorno precedente come variabili di stato, il Q-learning si è dimostrato più performante dell'SDP in cui l'afflusso è stato modellizzato come un processo autoregressivo di ordine uno. I metodi basati su RL sono in grado di alleviare in qualche modo anche la *Curse of Dimensionality*, in quanto lo spazio in cui si ricercano le decisioni ammissibili di rilascio non viene esplorato in maniera esaustiva ad ogni passo di iterazione. Ciò nonostante, come la SDP, questi metodi richiedono comunque una griglia di discretizzazione sullo spazio degli stati, la qual cosa porta di nuovo ad un'esplosione esponenziale dei costi di calcolo.

In seguito, è stato sviluppato un nuovo approccio, chiamato *fitted Q-iteration*, che combina i concetti di apprendimento fuori linea tipico del RL con l'approssimazione della funzione valore (Ernst *et al.*, 2005). Diversamente dagli algoritmi tradizionali di approssimazione stocastica (Bellman *et al.*, 1963; Bertsekas e Tsitsiklis, 1996), che utilizzano approssimatori di funzione parametrici, e che quindi richiedono un lungo e oneroso processo di stima parametrica ad ogni passo dell'iterazione, il fitted Q-iteration utilizza un'approssimazione tree-based (Breiman *et al.*, 1984). L'uso dei regressori tree-based offre un doppio vantaggio: in primo luogo, un'elevata flessibilità di modellizzazione, caratteristica fondamentale nel tipico contesto multi-obiettivo dei sistemi idrici con vettore di stato a più dimensioni, dove le funzioni valore da approssimare hanno una forma non direttamente riconoscibile; in secondo luogo, un'elevata efficienza di calcolo, poiché non è richiesta alcuna stima parametrica per l'approssimazione della funzione valore. Dall'altro lato, anche se i metodi tree-based ricavano la struttura del modello direttamente dai dati, alcuni parametri devono comunque essere specificati per guidare il processo di generazione degli alberi, come ad esempio il numero minimo di dati per foglia, oppure il numero di alberi. La scelta del valore di questi parametri può essere effettuata solamente per via empirica, e richiede un'accurata analisi ad hoc, in quanto ogni inesattezza può portare ad effetti negativi sulla prestazione della politica. Inoltre, mentre il Q-learning tradizionale converge solamente quando gli aggiornamenti della funzione valore sono realizzati in maniera incrementale, seguendo la traiettoria dello stato prodotta dalla sequenza delle decisioni ottime selezionate ad ogni passo dell'iterazione, il fitted Q-iteration processa l'informazione disponibile in modo batch, utilizzando simultaneamente tut-

ta l'esperienza di apprendimento per aggiornare la funzione valore.
Così come proposto in origine da Ernst *et al.*, (2005]), il fitted Q-iteration produce una politica stazionaria, che si adatta alla perfezione ai sistemi artificiali per i quali è stato sviluppato l'algoritmo, mentre si adatta molto meno ai sistemi naturali. Una versione migliorata che include politiche non stazionarie, più efficaci nell'adattarsi alla variabilità stagionale dei sistemi ambientali, è comunque disponibile, ed è quella utilizzata nel presente lavoro.

B.2 I limiti della SDP

Si consideri un generico problema di controllo ottimo stocastico e tempo-discreto (vedi Soncini-Sessa *et al.*, 2007 per maggiori dettagli):

$$p_H^* = \arg\max_{p_H} \mathrm{E}_{\boldsymbol{\varepsilon}_1,\ldots,\boldsymbol{\varepsilon}_h}\left[\sum_{t=0}^{H-1}\gamma^t g_t(\mathbf{X}_t,\mathbf{u}_t,\boldsymbol{\varepsilon}_{t+1})\right] \tag{B.1a}$$

$$\mathbf{X}_{t+1} = f_t(\mathbf{X}_t,\mathbf{u}_t,\boldsymbol{\varepsilon}_{t+1}) \qquad t=0,\ldots,H-1 \tag{B.1b}$$

$$m_t(\mathbf{X}_t) = \mathbf{u}_t \in \mathcal{U}_t(\mathbf{X}_t) \qquad t=0,\ldots,H-1 \tag{B.1c}$$

$$\boldsymbol{\varepsilon}_{t+1} \sim \phi_t(\cdot|\mathbf{X}_t,\mathbf{u}_t) \qquad t=0,\ldots,H-1 \tag{B.1d}$$

$$\mathbf{X}_0 \quad \text{dato} \tag{B.1e}$$

$$p_H \triangleq \{m_t(\cdot); t=0,\ldots,H-1\} \tag{B.1f}$$

La formulazione del problema di controllo ottimo (B.1) include un'ipotesi fondamentale della Programmazione Dinamica Stocastica (SDP): deve essere disponibile un modello esplicito del sistema, attraverso il quale possono essere anticipati gli effetti di ogni transizione di stato (*Curse of Modelling*). Precisamente:

1. Tutte le dinamiche del sistema sono note e devono essere esplicitamente modellizzate nell'equazione (B.1b); ciò sta a significare che l'informazione meteorologica e/o idrologica $\mathbf{W}_t$ può essere inclusa nella formulazione SDP come una variabile di stato descritta da un modello appropriato. Non è possibile considerare input esogeni deterministici, i cui valori sono noti in tempo reale (come la precipitazione, o la temperatura): gli input del modello possono essere solamente decisioni di rilascio o disturbi stocastici.

2. Il vettore dei disturbi è noto (equazione (B.1d)), e i disturbi sono indipendenti nel tempo, oppure.

3. Le funzioni di costo per passo sono note e separabili, ovvero $g_t(\cdot)$ dipende solamente da variabili definite nell'intervallo $[t, t+1)$.

La soluzione del problema (B.1) è calcolata risolvendo ricorsivamente la seguente equazione di Bellman nella formulazione TDC (vedi Soncini-Sessa *et al.*, 2007):

$$Q_t(\mathbf{X}_t, \mathbf{u}_t) = \mathrm{E}_{\boldsymbol{\varepsilon}_{t+1}}\Big[g_t(\mathbf{X}_t, \mathbf{u}_t, \boldsymbol{\varepsilon}_{t+1}) + \gamma \max_{\mathbf{u}_{t+1}} Q_{t+1}(\mathbf{X}_{t+1}, \mathbf{u}_{t+1})\Big] \quad \forall (\mathbf{X}_t, \mathbf{u}_t) \in \mathcal{S}_{X_t} \times \mathcal{S}_{\mathbf{u}_t} \tag{B.2}$$

dove $Q_t(\cdot)$ è denominata *Q-function* o *funzione valore*, ovvero il costo cumulato atteso risultante dall'applicazione della decisione di rilascio $\mathbf{u}_t$ al tempo t in corrispondenza dello stato $\mathbf{X}_t$ e dall'adozione di decisioni ottime in ognuna delle successive transizioni di stato (politica greedy). La relazione tra la *Q-function* e la funzione di costo atteso $H_t(\cdot)$, così come introdotta in origine da Bellman (1957), è data dalla seguente formula:

$$H_t(\mathbf{X}_t) = \max_{\mathbf{u}_t} Q_t(\mathbf{X}_t, \mathbf{u}_t) \tag{B.3}$$

che esprime lo stesso concetto in una notazione decisamente più compatta della precedente, ma che richiede un modello esplicito per ricavare la decisione di rilascio ottima associata ad ogni valore dello stato, in accordo con i requisiti della SDP. Più precisamente, la soluzione del problema (B.1) si ottiene risolvendo iterativamente l'equazione (B.2) con un processo risolutivo *backward-looking* sul periodo $T-1, \ldots, 0$ e ripetendo il ciclo finché viene soddisfatto un certo criterio di terminazione, ad esempio dopo un certo numero prefissato di cicli k. Infine, le ultime funzioni Q rappresentano le funzioni Q^* ottime, dalle quali si ricaverà la legge di controllo ottima per ogni t:

$$m_t^*(\mathbf{X}_t) = \arg\max_{\mathbf{u}_t} Q_t^*(\mathbf{X}_t, \mathbf{u}_t) \tag{B.4}$$

Per determinare la parte a destra dell'equazione (B.2), i domini $\mathcal{S}_{x_t}$, $\mathcal{S}_{u_t}$ e $\mathcal{S}_{\varepsilon_{t+1}}$, rispettivamente dello stato, del controllo e del disturbo, devono essere discretizzati e, ad ogni passo dell'iterazione del processo risolutivo, devono essere esplorati in maniera esaustiva. La scelta della discretizzazione del dominio è un'operazione essenziale, poiché si riflette sulla complessità dell'algoritmo, il quale è combinatorio rispetto al numero degli stati, dei controlli e delle decisioni, e nei lori domini di discretizzazione. Siano N_x, N_u e N_{ε_t} il numero di elementi di stato, controllo e disturbo discretizzati,

con $\mathcal{S}_{x_t} \in \mathbb{R}^{n_x}$, $\mathcal{S}_{u_t} \in \mathbb{R}^{n_u}$ e $\mathcal{S}_{\varepsilon_t} \in \mathbb{R}^{n_\varepsilon}$: la soluzione ricorsiva della (B.2) per kT passi d'iterazione (dove k è generalmente più piccolo di 10) richiede:

$$kT \cdot \left(N_{x_t}^{n_x} \cdot N_{u_t}^{n_u} \cdot N_{\varepsilon_t}^{n_\varepsilon}\right) \tag{B.5}$$

valutazioni dell'operatore E[·] nell'equazione (B.2). L'equazione (B.5) mostra esplicitamente l'effetto della *Curse of Dimensionality*, ovvero la crescita esponenziale della complessità di calcolo in funzione della dimensionalità dello stato e del controllo. Di conseguenza, la SDP non può essere utilizzata nel progetto di politiche di gestione giornaliere per sistemi idrici con un numero di serbatoi che eccede le poche unità, in genere 2 o 3, e/o quando si considerano troppe variabili idro-meteorologiche nel vettore $\mathbf{W}_t$.

B.3 Tree-based batch mode Reinforcement Learning

Come anticipato, il Reinforcement Learning (RL) fornisce una procedura concettuale in grado di superare la *Curse of Modelling*, in quanto non richiede la conoscenza di un modello esplicito per descrivere le transizioni di stato, le densità di probabilità del disturbo e i costi. Tuttavia, riesce ad alleviare solo in parte la *Curse of Dimensionality* espressa dall'equazione (B.5).

L'algoritmo di fitted Q-iteration (Ernst *et al.*, 2005), combina l'idea di apprendimento dall'esperienza tipica del RL con il concetto di approssimazione continua della funzione valore sviluppato per la programmazione dinamica su larga scala. Tutto questo si traduce in un'accentuata riduzione dell'onere computazionale. Infatti, una mappatura continua della coppia stato-decisione nella funzione valore dovrebbe permettere lo stesso livello di accuratezza di una rappresentazione tabellare (look-up table) basata su di una griglia estremamente densa, utilizzando però una griglia molto più lasca per lo spazio stato-decisione. Inoltre, il processo di apprendimento è realizzato fuori linea, senza la necessità di sperimentare direttamente sul sistema reale: questo è un requisito fondamentale quando si ha a che fare con sistemi di risorse idriche, in quanto gli esperimenti sul sistema reale porterebbero a costi insostenibili in termini di tempo e di perdite sociali ed economiche [Soncini-Sessa *et al.*, 2007].

B.3.1 L'algoritmo di fitted Q-iteration

Analogamente agli altri algoritmi di apprendimento per rinforzo, il fitted Q-iteration non richiede la modellizzazione esplicita del sistema. La politica di controllo è determinata dall'apprendimento dall'esperienza. Più precisamente, tale esperienza è rappresentata da un dataset finito $\mathcal{T}$ composto da tuple della forma $< \mathbf{x}_t, \mathbf{u}_t, \mathbf{x}_{t+1}, g_{t+1} >$, ovvero

$$\mathcal{T} = \{< \mathbf{x}_t^l, \mathbf{u}_t^l, \mathbf{x}_{t+1}^l, g_{t+1}^l >, l = 1, \ldots, \#\mathcal{F}\}$$

dove $\#\mathcal{T}$ rappresenta la cardinalità di $\mathcal{T}$. Ogni tupla è un esempio di transizione di un passo delle dinamiche del sistema. L'insieme $\mathcal{T}$ è l'unica informazione richiesta per determinare una politica di gestione, indipendentemente dal modo in cui viene generata (vedi par. B.3.2), sia che le tuple siano ottenute da una singola traiettoria del sistema (ad esempio, la traiettoria storica), sia da diverse simulazioni ad un passo o multi-step, generate indipendentemente, delle dinamiche del sistema. Siccome, a parte casi molto rari, una politica ottima non può essere determinata a partire da un insieme finito di transizioni, la politica generata dal fitted Q-batch sarà un'approssimazione della politica ottima p^* che risolve il problema (B.1).
Il caso deterministico stazionario è utile per descrivere il funzionamento dell'algoritmo. Sotto queste ipotesi semplificative, la transizione di stato (B.1b) e il costo associato dipendono solamente dallo stato $\mathbf{x}_t$ e dalla decisione $\mathbf{u}_t$. Si può dimostrare (Ernst, 1999) che la seguente sequenza di funzioni Q_h, definite per tutte le coppie $(\mathbf{x}_t, \mathbf{u}_t) \in \mathcal{S}_x \times \mathcal{S}_u$

$$Q_0(\mathbf{x}_t, \mathbf{u}_t) = 0 \tag{B.6a}$$

$$Q_h(\mathbf{x}_t, \mathbf{u}_t) = g(\mathbf{x}_t, \mathbf{u}_t) + \gamma \max_{\mathbf{u}_{t+1}} Q_{h-1}(\mathbf{x}_{t+1}, \mathbf{u}_{t+1}) \quad \forall h > 0 \tag{B.6b}$$

converge alla funzione Q ottima $Q^*(\cdot)$ che risolve il problema deterministico e stazionario equivalente all'equazione (B.2). Ipotizzando che la funzione $Q_{h-1}(\cdot)$ sia nota, il valore di $Q_h(\cdot)$ può essere calcolato per tutte le coppie stato-decisione $(\mathbf{x}_t^l, \mathbf{u}_t^l), l = 1, \ldots, \#\mathcal{T}$, utilizzando l'equazione (B.6b) e il dataset di apprendimento $\mathcal{T}$. Il valore della cardinalità di $\mathcal{T}$ così ottenuto può essere utilizzato in seguito per ottenere un'approssimazione continua $\hat{Q}_h(\cdot)$ di $Q_h(\cdot)$ sull'intero insieme stato-decisione $\mathcal{S}_x \times \mathcal{S}_u$ applicando un algoritmo di regressione al set di addestramento

$$\mathcal{TS} = \{< (\mathbf{x}_t^l, \mathbf{u}_t^l), Q_H(\mathbf{x}_t^l, \mathbf{u}_t) >, l = 1, \ldots, \#\mathcal{T}\}$$

dove le coppie $(\mathbf{x}_t^l, \mathbf{u}_t^l)$ sono gli input del regressore e i valori $Q_h(\mathbf{x}_t^l, \mathbf{u}_t)$ sono gli output del regressore. Sostituendo $Q_h(\cdot)$ con $\hat{Q}_h(\cdot)$ e applicando lo stesso ragionamento, le approssimazioni successive $\hat{Q}_{h+1}(\cdot), \hat{Q}_{h+2}(\cdot), \ldots$ possono essere determinate in maniera iterativa.
Nel caso stocastico, la parte a destra dell'equazione (B.6b) è una realizzazione di una variabile casuale e $Q_h(\mathbf{x}_t, \mathbf{u}_t)$ è ridefinita secondo questo concetto. Tuttavia, il valore atteso non deve essere applicato in via operativa quando $\hat{Q}_h(\cdot)$ è approssimata con un regressore basato sui minimi quadrati, in quanto quest'ultimo genera un'approssimazione del valore atteso dell'output condizionato all'input. La sua applicazione al set di apprendimento costruito considerando le transizioni stocastiche genera un'approssimazione continua di $Q_h(\cdot)$ sull'intero insieme stato-decisione.
Così come proposto in origine da Ernst *et al.*, 2005, l'algoritmo di fitted Q-iteration genera una politica stazionaria, ovvero un'unica legge di controllo della forma $\mathbf{u}_t = m(\mathbf{x}_t)$, che rappresenta la politica ottima per un sistema stazionario. Tuttavia, i sistemi naturali non sono affatto stazionari e quindi una politica periodica è molto più indicata per adattarsi alla variabilità stagionale. Un modo per estendere la procedura di fitted Q-iteration al caso non stazionario è quello di considerare il tempo come un componente del vettore di stato, che evolve guidato da una funzione di transizione di stato autonoma e deterministica: t è sempre seguito da $t+1$. Detto questo, la notazione del dataset di apprendimento $\mathcal{T}$ può essere riscritta come

$$\mathcal{T} = \{< (t, \mathbf{x}_t)^l, \mathbf{u}_t^l, (t+1, \mathbf{x}_{t+1})^l, g_{t+1}^l >, l = 1, \ldots, \#\mathcal{T}\}$$

In questo modo, tutte le proprietà della formulazione stazionaria sono conservate e la convergenza può essere dimostrata sotto le stesse ipotesi del caso precedente.
Una versione schematica dell'algoritmo di fitted Q-iteration modificato in questo modo è il seguente:

Input: un dataset $\mathcal{T}$ di apprendimento e un algoritmo di regressione.

Inizializzazione:

Porre $h = 0$.

Porre $\hat{Q}_0(\cdot) = 0$ sull'intero spazio stato-decisione $\mathcal{S}_x \times \mathcal{S}_u$.

Iterazioni: ripetere finché la condizione di terminazione non è rispettata

Porre $h = h + 1$.

Costruire il set di addestramento $\mathcal{TS} = \{< i^l, o^l >, l = 1, \ldots, \#T\}$ dove $i^l = ((t, \mathbf{x}_t)^l$ e $o^l = g^l_{t+1} + \gamma \max_{u_{t+1}} \hat{Q}_{H-1}((t+1, \mathbf{x}_{t+1})^l, \mathbf{u}_{t+1})$

Applicare l'algoritmo di regressione su TS per ottenere $\hat{Q}_h(\cdot)$, dalla quale deriva la politica $\hat{p}_h$.

L'algoritmo di fitted Q-iteration è un algoritmo batch, in quanto l'intero dataset di apprendimento $\mathcal{T}$ è processato in modo batch, al contrario degli algoritmi RL tradizionali che operano un aggiornamento incrementale della funzione valore utilizzando le tuple in maniera sequenziale. Le iterazioni possono essere fermate quando la differenza tra $\hat{Q}_h(\cdot)$ e $\hat{Q}_{h-1}(\cdot)$ scende al di sotto di una soglia prefissata, anche se questo criterio non assicura la convergenza con alcuni approssimatori di funzione (vedi par. B.3.3). Quando l'algoritmo termina, qualsiasi sia la condizione di terminazione selezionata, la politica finale $\hat{p}$ è un'approssimazione della politica ottima p^*. La politica $\hat{p}_h$ associata all'h-esima iterazione del'algoritmo è composta da una sequenza di $T-1$ leggi di controllo, ognuna delle quali insiste sull'orizzonte $[t, t+h)$. In altre parole, per ogni valore di h, l'algoritmo risolve un problema ricorsivo con passo h della forma (B.1).

B.3.2 Il dataset di apprendimento

In accordo con il concetto di apprendimento fuori linea dall'esperienza tipico del RL, l'idea più semplice per generare il dataset di apprendimento $\mathcal{T}$ è quello di impiegare una serie storica di transizioni del sistema e lasciare che l'algoritmo impari dall'esperienza reale. Se gli obiettivi selezionati per il problema si adattano bene ai target attuali di gestione, la politica così derivata sarà molto simile a quella storica, con benefici molto piccoli nel caso in cui il sistema sia attualmente gestito al di sotto del suo potenziale. Un modo per raffinare e migliorare questa politica quasi-storica è quello di allargare l'esplorazione delle decisioni di rilascio ad un piccolo insieme di valori diversi attorno a quelli storici (vedi Gaskett, 2002), per ognuno dei valori passati dello stato. Tuttavia, si tratta di un approccio rischioso: se il set spazio-decisione è stato scarsamente rilevato durante la gestione storica, il contenuto

informativo del dataset di apprendimento può essere basso, e la politica di gestione risultante potrebbe essere con tutta probabilità molto lontana dall'ottimo. Inoltre, l'approccio è di per sè impraticabile quando il sistema idrico non è mai stato gestito in passato (ad esempio, in problemi di pianificazione).
Un approccio alternativo è quello di studiare il comportamento del sistema idrico, tramite simulazione, per differenti valori dello stato e per differenti politiche di controllo, in altre parole adottare un approccio model-based. Tuttavia, lo sforzo modellistico non deve necessariamente coinvolgere l'intero sistema idrico, ma solamente le componenti direttamente controllate (cioè i serbatoi) e ogni componente di valle che sia influenzata dalle decisioni di rilascio (ad esempio, gli utenti di valle). Infatti, la parte a monte del sistema (cioè i sistemi meteo/bacino) non è influenzata dalle decisioni di rilascio, perciò non è richiesto un modello per esplorarne le dinamiche dei processi e le realizzazioni dei disturbi. Questa è l'idea di fondo dell'approccio parzialmente model-free (Castelletti *et al.*, 2001): utilizzare le serie temporali storicamente rilevate per ognuno degli afflussi e, quando disponibili, per ognuna delle altre informazioni idrometeorologiche che possono essere incluse all'interno delle variabili di stato, descrivendo invece le dinamiche dell'invaso con semplici equazioni di bilancio di massa, e descrivendo qualsiasi utilizzatore di valle con un appropriato modello dinamico (vedi, come esempio, Galelli *et al.*, 2010). Finché si considerano solamente le parti direttamente controllate, è richiesta la discretizzazione dei corrispondenti spazi di stato e di decisione, per poter eseguire le simulazioni ad un passo o multi-step delle dinamiche rilevanti. Anche se scremata dalle componenti d'informazione idrometeorologica, la discretizzazione a griglia densa adottata nella formulazione SDP può comunque portare a richieste computazionali proibitive. Dall'altro lato, non si riuscirebbe a ricavare alcun vantaggio dall'approssimazione continua delle funzioni Q data dall'algoritmo di fitted Q-iteration. Diversamente, una griglia lasca può ridurre esponenzialmente l'onere computazionale riducendo linearmente N_x e N_u nell'equazione (B.5). Una griglia lasca può essere ottenuta sia con un sotto-campionamento uniforme della griglia densa della SDP o generata con un metodo di discretizzazione più efficiente, come gli *orthogonal arrays*, *Latin hypercube designs*, e *low discrepancy sequences* (vedi Cervellera *et al.*, 2006). Qualsiasi sia l'approccio adottato per costruire il dataset di apprendimento, quest'ultimo può contenere le ridondanze entro un margine accettabile: tale proprietà va ad aggiungersi solamente ai requisiti computazionali, senza alcun vantaggio in termini di prestazione della politica. Un modo per filtrare il dataset è quello di adottare

tecniche di apprendimento attivo (Cohn *et al.* 1996), secondo le quali vengono mantenute solo i campioni che migliorano maggiormente le prestazioni dell'algoritmo di apprendimento (vedi Ernst, 2005).

B.3.3 L'approssimatore di funzione

In via di principio, l'algoritmo di fitted Q-iteration può essere combinato con qualsiasi approssimatore di funzioni basato sui minimi quadrati e progettato per problemi di regressione. In pratica, l'approssimatore adottato dovrebbe possedere due caratteristiche:

1. *Flessibilità.* Per problemi molto semplici, che riguardano, ad esempio, un singolo serbatoio gestito con un unico obiettivo, la classe di funzioni, alla quali appartengono le funzioni Q da approssimare, può essere riconosciuta a priori, entro certi limiti (ad esempio per il controllo delle piene, deve essere una funzione monotona crescente con l'invaso). Tuttavia, quando si ha a che fare con reti di serbatoi e/o problemi multi-obiettivo, la forma della funzione può essere completamente imprevedibile. Di conseguenza, l'approssimatore di funzione deve essere in grado di adattarsi alla struttura del problema.

2. *Efficienza computazionale.* L'algoritmo di regressione viene eseguito ad ogni passo d'iterazione del fitted Q-iteration. Deve perciò assicurare approssimazioni accurate, senza aggiungere troppo ai requisiti di calcolo totali.

Alcuni approssimatori parametrici di funzione riescono a garantire una grande flessibilità di modellizzazione; le reti neurali artificiali, ad esempio, sono in grado di approssimare qualsiasi funzione continua a più variabili a qualunque grado di accuratezza desiderato. Questa flessibilità, tuttavia, ha un prezzo, in quanto si riflette spesso in un elevato numero di parametri che richiedono una calibrazione esplicita, e quindi vanno ad influire sull'efficienza di calcolo (vedi Castelletti *et al.*, 2005) e aumentano il rischio di sovra-parametrizzazione. All'aumentare della grandezza del problema, le reti neurali richiedono sempre più neuroni, incrementando perciò il costo in termini di potenza di calcolo della fase di addestramento della politica. Gli approssimatori di funzione non parametrici, in particolare i metodi tree-based, assicurano flessibilità di modellizzazione e, allo stesso tempo, efficienza computazionale, poiché non richiedono una stima parametrica tradizionale nel loro processo di costruzione.

I metodi tree-based forniscono stime non parametriche basate su una divisione binaria ricorsiva del dataset di addestramento $\mathcal{TS}$ (algoritmo di costruzione dell'albero). Al primo passaggio, lo spazio degli input (o radice) è partizionato in due sottoinsiemi (nodi), applicando a $\mathcal{TS}$ una *splitting rule* appropriata. L'operazione viene ripetuta in via iterativa sui due sottoinsiemi risultanti da ogni divisione, fino a che una data condizione di terminazione risulta essere soddisfatta. Ad ogni sottoinsieme della partizione finale (foglia) viene associato un valore dell'output o una funzione dell'input (regola di associazione). In alcuni metodi, il processo di costruzione dell'albero viene ripetuto più volte, per costruire una foresta di alberi, e i valori stimati tramite gli alberi vengono aggregati allo scopo di produrre la stima finale (per maggiori dettagli sul funzionamento dei metodi tree-based, vedi App. A).

Appendice C

Risultati fase di riduzione

C.1 Temperatura di valle g_t^{temp}

Tabella C.1: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a g_t^{temp}.

Passo 0

Variabile di output	g_t^{temp}
Varianza variabile di output	$8.3762 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	5.3329	446698	0.0297
u_t^{-3}	5.1085	427897	0.0057
u_t^{-13}	4.9112	411367	0.0057
$hTaff_t$	4.0181	336566	0.0408
T_t^K	3.9953	334655	0.1139

Passo 1

Variabile di output	$g_t^{temp} - \hat{r}_t^0$
Varianza variabile di output	$7.4235 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	6.7224	499038	0.0103
u_t^{-3}	6.5719	487865	0.0061
u_t^{-13}	6.3781	473478	0.0058
$hTaff_t$	5.2379	388836	0.0175
T_t^{-3}	2.1825	162017	-0.0025

Passo 1.1

Variabile di output	$g_t^{temp} - \hat{r}_t^1$
Varianza variabile di output	$7.3419 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	5.9096	433876	0.0115
u_t^{-13}	5.8698	430958	0.0054
u_t^{-3}	5.8384	428649	0.0053
$Taff_t$	1.9573	143702	0.0026
$t_{mod\ T}$	1.9118	140359	0.0091

Passo 1.2

Variabile di output	$g_t^{temp} - \hat{r}_t^2$
Varianza variabile di output	$6.6653 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	7.5981	506442	0.0054
u_t^{-13}	7.54425	502850	0.0055
$NH_{4,t}^K$	1.9296	128612	0.0209
T_t^{-3}	1.9100	127310	-0.0031
$NH_{4,t}^F$	1.6388	109232	0.0209

Passo 1.3

Variabile di output	$g_t^{temp} - \hat{r}_t^3$
Varianza variabile di output	$6.0411 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	12.3908	748544	0.0061
u_t^{-13}	12.1586	734514	0.0054
T_t^{-3}	3.9488	238553	-0.0034
T_t^{-7}	2.6586	160611	-0.0033
T_t^{bot}	1.7821	107659	-0.0056

Passo 1.4

Variabile di output	$g_t^{temp} - \hat{r}_t^4$
Varianza variabile di output	$5.5713 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	13.8558	771947	0.0149
T_t^{-3}	4.9894	277975	-0.0034
T_t^{-7}	3.3983	189331	-0.0036
T_t^{-13}	1.6902	94164.2	-0.0049
h_t	1.4946	83267.6	-0.0052

Passo 1.5

Variabile di output	$g_t^{temp} - \hat{r}_t^5$
Varianza variabile di output	$2.9831 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.2563	67308.6	-0.0032
h_t	2.2303	66531.8	-0.0045
s_t	1.9761	58949.1	-0.0035
$t_{mod\ T}$	1.9557	58341.6	0.0045
T_t^{-13}	1.6294	48607	-0.0059

Passo 1.6

Variabile di output	$g_t^{temp} - \hat{r}_t^6$
Varianza variabile di output	$2.9567 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.3422	69250.8	-0.0032
T_t^{bot}	2.2970	67916.1	-0.0046
s_t	1.7385	51402.2	-0.0026
T_t^{-13}	1.6853	49828.6	-0.0048
T_t^{-7}	1.5605	46138.4	-0.0019

Passo 1.7

Variabile di output	$g_t^{temp} - \hat{r}_t^7$
Varianza variabile di output	$2.8462 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.2323	63535.1	-0.0048
h_t	2.1576	61410	-0.0028
s_t	1.8751	53367.3	-0.0026
T_t^{-13}	1.6064	45720.7	-0.0043
$Cloud - Cover_t$	1.2895	36700.3	0.0005

C.2 Clorofilla nel serbatoio g_t^{rec}

Tabella C.2: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a g_t^{rec}.

Passo 0

Variabile di output	g_t^{rec}
Varianza variabile di output	736913

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
DO_t^K	9.1436	67380.7	0.6124
T_t^K	9.0528	66711.6	0.6121
T_t^K	8.4943	62595.3	0.6021
$Air - Temp_t$	7.3721	54325.7	0.5933
DO_t^F	7.1703	52838.7	0.6026

Passo 1

Variabile di output	$g_t^{rec} - \hat{r}_t^0$
Varianza variabile di output	285165

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	7.7155	22001.9	0.0720
h_t	6.7304	19192.8	0.1244
s_t	4.9288	14055.2	0.1163
$NO_{3,t}^F$	2.7809	7930.04	0.2818
$hTaff_t$	2.7436	7823.82	0.0788

Passo 1.1

Variabile di output	$g_t^{rec} - \hat{r}_t^1$
Varianza variabile di output	175416

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	11.8124	20720.8	0.1489
s_t	8.9162	15640.3	0.1486
T_t^{sed}	5.3075	9310.26	0.1114
$hTaff_t$	4.4164	7747.09	0.0888
T_t^{-3}	3.7702	6613.58	0.0232

Passo 1.2

Variabile di output	$g_t^{rec} - \hat{r}_t^2$
Varianza variabile di output	129306

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	9.4999	12284	0.0841
$t_{mod\ T}$	6.0980	7885.09	0.1065
$hTaff_t$	4.8920	6325.69	0.0173
$hTSSmax_t$	3.0586	3954.9	0.0225
T_t^{-3}	2.8972	3746.25	0.0063

Passo 1.3

Variabile di output	$g_t^{rec} - \hat{r}_t^3$
Varianza variabile di output	78359.9

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	5.9401	4654.65	0.0207
$hTaff_t$	4.4490	3486.26	0.0038
T_t^{-3}	2.2315	1748.6	0.0041
$Cloud - Cover_t$	2.1871	1713.78	0.0763
T_t^{-7}	1.9284	1511.12	0.0058

Passo 1.4

Variabile di output	$g_t^{rec} - \hat{r}_t^4$
Varianza variabile di output	74178.4

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	6.0668	4500.27	0.0201
$hTaff_t$	4.7929	3555.27	0.0041
T_t^{-3}	2.3762	1762.63	0.0035
T_t^{bot}	2.1189	1571.78	0.0092
T_t^{-7}	2.0458	1517.56	0.0059

C.3 Sedimentazione nel serbatoio g_t^{sed}

Tabella C.3: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a g_t^{sed}.

Passo 0

Variabile di output	g_t^{sed}
Varianza variabile di output	$4.3720 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$SSOL6_t^K$	8.4920	$3.7127 \cdot 10^{20}$	0.4385
$SSOL3_t^K$	8.2978	$3.6279 \cdot 10^{20}$	0.4475
$SSOL3_t^F$	7.5080	$3.2825 \cdot 10^{20}$	0.4376
$SSOL4_t^K$	7.1152	$3.1108 \cdot 10^{20}$	0.4557
$SSOL5_t^K$	7.1147	$3.1106 \cdot 10^{20}$	0.4896

Passo 1

Variabile di output	$g_t^{sed} - \hat{r}_t^0$
Varianza variabile di output	$2.3832 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$SSOL2_t^K$	6.8297	$1.6277 \cdot 10^{19}$	0.4569
$SSOL1_t^K$	6.0372	$1.4388 \cdot 10^{19}$	0.4659
$SSOL4_t^K$	5.9398	$1.4156 \cdot 10^{19}$	0.4553
$SSOL6_t^F$	5.5812	$1.3301 \cdot 10^{19}$	0.4524
$SSOL3_t^F$	5.5280	$1.3175 \cdot 10^{19}$	0.4693

Passo 1.1

Variabile di output	$g_t^{sed} - \hat{r}_t^1$
Varianza variabile di output	$2.2293 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$SSOL6_t^F$	5.3815	$6.6157 \cdot 10^{18}$	0.2433
$SSOL1_t^F$	5.0834	$6.2491 \cdot 10^{18}$	0.2460
$SSOL4_t^K$	4.9811	$6.1234 \cdot 10^{18}$	0.2355
$SSOL4_t^F$	4.9230	$6.0520 \cdot 10^{18}$	0.2472
$SSOL5_t^F$	4.7617	$5.8538 \cdot 10^{18}$	0.2342

Passo 1.2

Variabile di output	$g_t^{sed} - \hat{r}_t^2$
Varianza variabile di output	$7.3972 \cdot 10^{19}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
TSS_t^{-3}	8.9740	$6.6382 \cdot 10^{18}$	0.2195
$POCL_t^K$	4.1540	$3.0728 \cdot 10^{18}$	0.1083
$SSOL5_t^F$	3.9796	$2.9438 \cdot 10^{18}$	0.0891
TSS_t^{-7}	3.6301	$2.6852 \cdot 10^{18}$	0.1834
$SSOL6_t^K$	3.5035	$2.5916 \cdot 10^{18}$	0.1018

Passo 1.3

Variabile di output	$g_t^{sed} - \hat{r}_t^3$
Varianza variabile di output	$6.1080 \cdot 10^{19}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$SSOL1_t^K$	4.8272	$2.9485 \cdot 10^{18}$	0.0081
$SSOL1_t^F$	4.1467	$2.5328 \cdot 10^{18}$	0.0072
$SSOL2_t^K$	3.9991	$2.4427 \cdot 10^{18}$	0.0174
$SSOL5_t^F$	3.8541	$2.3541 \cdot 10^{18}$	-0.0166
$SSOL3_t^K$	3.7750	$2.3058 \cdot 10^{18}$	0.0003

Passo 1.4

Variabile di output	$g_t^{sed} - \hat{r}_t^4$
Varianza variabile di output	$4.07254 \cdot 10^{19}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$SSOL6_t^K$	3.2720	$1.3325 \cdot 10^{18}$	-0.0246
$SSOL5_t^F$	2.8546	$1.1626 \cdot 10^{18}$	-0.0072
$SSOL1_t^F$	2.8335	$1.1539 \cdot 10^{18}$	-0.0106
$SSOL1_t^K$	2.7875	$1.1352 \cdot 10^{18}$	-0.0087
$SSOL4_t^K$	2.7785	$1.1315 \cdot 10^{18}$	0.0145

C.4 Deficit irriguo g_t^{irr}

Tabella C.4: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a g_t^{irr}.

Passo 0

Variabile di output	g_t^{irr}
Varianza variabile di output	$1.8121 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	18.9908	$3.4414 \cdot 10^{20}$	0.0277
u_t^{-13}	18.9258	$3.4296 \cdot 10^{20}$	0.0329
$t_{mod\ T}$	8.8469	$1.6032 \cdot 10^{20}$	0.0516
T_t^K	2.4815	$4.4968 \cdot 10^{19}$	0.1273
T_t^F	2.3160	$4.1969 \cdot 10^{19}$	0.1384

Passo 1

Variabile di output	$g_t^{irr} - \hat{r}_t^0$
Varianza variabile di output	$1.5068 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	16.9516	$2.5543 \cdot 10^{20}$	0.0368
u_t^{-3}	16.4875	$2.4843 \cdot 10^{20}$	0.0331
$t_{mod\ T}$	10.4201	$1.5701 \cdot 10^{20}$	-0.0417
T_t^K	3.5363	$5.3285 \cdot 10^{19}$	-0.0397
$Air - Temp_t$	2.6866	$4.0482 \cdot 10^{19}$	-0.0380

Passo 1.1

Variabile di output	$g_t^{irr} - \hat{r}_t^1$
Varianza variabile di output	$1.3684 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	16.4875	$2.4843 \cdot 10^{20}$	0.0187
$t_{mod\ T}$	10.4201	$1.5701 \cdot 10^{20}$	0.0405
T_t^K	3.5363	$5.3285 \cdot 10^{19}$	0.0357
DO_t^K	2.6866	$4.0482 \cdot 10^{19}$	0.0432
$Air - Temp_t$	2.6317	3.965510^{19}	0.0495

Passo 1.2

Variabile di output	$g_t^{irr} - \hat{r}_t^2$
Varianza variabile di output	$1.3284 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	9.1731	$1.2186 \cdot 10^{20}$	0.0182
$t_{mod\ T}$	8.9150	$1.1843 \cdot 10^{20}$	0.0320
T_t^K	2.9374	$3.9021 \cdot 10^{19}$	0.0290
DO_t^K	2.3867	$3.1706 \cdot 10^{19}$	0.0337
DO_t^F	2.0041	$2.6623 \cdot 10^{19}$	0.0397

Passo 1.3

Variabile di output	$g_t^{irr} - \hat{r}_t^3$
Varianza variabile di output	$1.3098 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	9.0532	$1.1858 \cdot 10^{20}$	0.0187
$t_{mod\ T}$	8.9853	$1.1769 \cdot 10^{20}$	0.0279
T_t^K	2.8259	$3.7013 \cdot 10^{19}$	0.0303
DO_t^K	2.5047	$3.2806 \cdot 10^{19}$	0.0272
$Vap - Press_t$	1.5983	$2.0934 \cdot 10^{19}$	0.0607

Passo 1.4

Variabile di output	$g_t^{irr} - \hat{r}_t^4$
Varianza variabile di output	$1.2741 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$t_{mod\ T}$	9.7059	$1.2366 \cdot 10^{20}$	0.0212
u_t^{-3}	9.1185	$1.1618 \cdot 10^{20}$	0.0183
T_t^K	2.9352	$3.7397 \cdot 10^{19}$	0.0201
DO_t^K	2.4395	$3.1082 \cdot 10^{19}$	0.0180
s_t	1.6515	$2.1042 \cdot 10^{19}$	-0.0113

Passo 1.5

Variabile di output	$g_t^{irr} - \hat{r}_t^5$
Varianza variabile di output	$1.1043 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	10.5067	$1.1602 \cdot 10^{20}$	0.0245
T_t^K	2.8938	$3.1955 \cdot 10^{19}$	0.0266
DO_t^K	2.3399	$2.5838 \cdot 10^{19}$	0.0265
s_t	1.9568	$2.1609 \cdot 10^{19}$	-0.0153
h_t	1.6679	$1.8418 \cdot 10^{19}$	-0.0006

Passo 1.6

Variabile di output	$g_t^{irr} - \hat{r}_t^6$
Varianza variabile di output	$1.1020 \cdot 10^{21}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	10.2677	$1.1315 \cdot 10^{20}$	0.0244
DO_t^K	2.3886	$2.6321 \cdot 10^{19}$	0.0220
s_t	1.9904	$2.1933 \cdot 10^{19}$	-0.0158
h_t	1.6485	$1.8166 \cdot 10^{19}$	-0.0010
T_t^{-3}	1.2299	$1.3553 \cdot 10^{19}$	0.0021

Passo 1.7

Variabile di output	$g_t^{irr} - \hat{r}_t^7$
Varianza variabile di output	$5.3968 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
DO_t^K	2.5244	$1.3624 \cdot 10^{19}$	0.0294
$NH_{4,t}^K$	2.3026	$1.2426 \cdot 10^{19}$	0.2513
$NH_{4,t}^F$	2.0110	$1.0853 \cdot 10^{19}$	0.2503
$NO_{3,t}^K$	1.9601	$1.0578 \cdot 10^{19}$	0.2621
TSS_t^{-3}	1.8661	$1.0071 \cdot 10^{19}$	0.0391

Passo 1.8

Variabile di output	$g_t^{irr} - \hat{r}_t^8$
Varianza variabile di output	$3.4776 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
SW_t	2.0865	$7.2559 \cdot 10^{18}$	0.0451
h_t	1.9533	$6.7929 \cdot 10^{18}$	0.0106
s_t	1.8646	$6.4842 \cdot 10^{18}$	0.0003
$Rain_t$	1.7686	$6.1504 \cdot 10^{18}$	0.0003
$hTaff_t$	1.3012	$4.5250 \cdot 10^{18}$	0.0007

Passo 1.9

Variabile di output	$g_t^{irr} - \hat{r}_t^9$
Varianza variabile di output	$3.4729 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.0765	$7.2116 \cdot 10^{18}$	0.0098
s_t	1.9465	$6.7602 \cdot 10^{18}$	-0.0001
$Rain_t$	1.7890	$6.2132 \cdot 10^{18}$	0.0002
$hTaff_t$	1.2906	$4.4821 \cdot 10^{18}$	0.0006
DO_t^K	1.1378	$3.9514 \cdot 10^{18}$	0.0151

Passo 1.10

Variabile di output	$g_t^{irr} - \hat{r}_t^{10}$
Varianza variabile di output	$3.4680 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.0422	$7.0825 \cdot 10^{18}$	0.0104
s_t	1.9306	$6.6954 \cdot 10^{18}$	-0.0002
$Rain_t$	1.6869	$5.8502 \cdot 10^{18}$	0.0001
$hTaff_t$	1.2616	$4.3751 \cdot 10^{18}$	0.0007
T_t^{bot}	1.0668	$3.6999 \cdot 10^{18}$	0.0076

Passo 1.11

Variabile di output	$g_t^{irr} - \hat{r}_t^{11}$
Varianza variabile di output	$2.8612 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$Rain_t$	2.0867	$5.9706 \cdot 10^{18}$	0.0001
$hTaff_t$	1.3473	$3.8549 \cdot 10^{18}$	-0.0006
s_t	1.2103	$3.4629 \cdot 10^{18}$	0.0003
$gateTSSmax_t$	0.9299	$2.6607 \cdot 10^{18}$	0.0025
$NH_{4,t}^K$	0.9226	$2.6399 \cdot 10^{18}$	0.0466

Passo 1.12

Variabile di output	$g_t^{irr} - \hat{r}_t^{12}$
Varianza variabile di output	$2.7531 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$Rain_t$	1.7698	$4.8725 \cdot 10^{18}$	0.0004
$hTaff_t$	1.4446	$3.9769 \cdot 10^{18}$	-0.0014
s_t	1.2934	$3.5608 \cdot 10^{18}$	-0.0006
$NH_{4,t}^F$	0.9075	$2.4984 \cdot 10^{18}$	0.0047
$gateTaff_t$	0.7581	$2.0870 \cdot 10^{18}$	-0.0002

Passo 1.13

Variabile di output	$g_t^{irr} - \hat{r}_t^{13}$
Varianza variabile di output	$2.7421 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$Rain_t$	1.6758	$4.5951 \cdot 10^{18}$	0.0004
$hTaff_t$	1.3411	$3.6774 \cdot 10^{18}$	-0.0014
s_t	1.3324	$3.6536 \cdot 10^{18}$	-0.0020
$gateTaff_t$	0.9070	$2.4870 \cdot 10^{18}$	-0.0003
$PO_{4,t}^F$	0.7465	$2.0469 \cdot 10^{18}$	0.0020

Passo 1.14

Variabile di output	$g_t^{irr} - \hat{r}_t^{14}$
Varianza variabile di output	$2.7543 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$Rain_t$	1.6600	$4.5723 \cdot 10^{18}$	0.0002
$hTaff_t$	1.4663	$4.0388 \cdot 10^{18}$	-0.0015
s_t	1.2859	$3.5419 \cdot 10^{18}$	-0.0015
$gateTaff_t$	0.8066	$2.2217 \cdot 10^{18}$	-0.0004
$PO_{4,t}^K$	0.7001	$1.9283 \cdot 10^{18}$	-0.0004

Passo 1.15

Variabile di output	$g_t^{irr} - \hat{r}_t^{15}$
Varianza variabile di output	$2.7049 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	1.4262	$3.8577 \cdot 10^{18}$	-0.0019
s_t	1.2116	$3.2772 \cdot 10^{18}$	-0.0022
$Taff_t$	0.7206	$1.9493 \cdot 10^{18}$	-0.0033
$PO_{4,t}^K$	0.7043	$1.9050 \cdot 10^{18}$	-0.0090
$gateTaff_t$	0.6827	$1.8466 \cdot 10^{18}$	-0.0004

Passo 1.16

Variabile di output	$g_t^{irr} - \hat{r}_t^{16}$
Varianza variabile di output	$2.6870 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	1.3548	$3.6403 \cdot 10^{18}$	-0.0023
s_t	1.2299	$3.3046 \cdot 10^{18}$	-0.0059
$Taff_t$	0.7177	$1.9285 \cdot 10^{18}$	-0.0098
$PO_{4,t}^K$	0.6809	$1.8295 \cdot 10^{18}$	-0.0110
$Wind - Speed_t$	0.6177	$1.6598 \cdot 10^{18}$	-0.0089

Passo 1.17

Variabile di output	$g_t^{irr} - \hat{r}_t^{17}$
Varianza variabile di output	$2.6720 \cdot 10^{20}$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	1.2611	$3.3696 \cdot 10^{18}$	-0.0056
$Taff_t$	0.7300	$1.9506 \cdot 10^{18}$	-0.0095
$PO_{4,t}^K$	0.6374	$1.7032 \cdot 10^{18}$	-0.0098
$Wind - Speed_t$	0.6207	$1.6585 \cdot 10^{18}$	-0.0064
$Cloud - Cover_t$	0.5829	$1.5576 \cdot 10^{18}$	-0.0008

C.5 Dinamica variabili di stato

C.5.1 Dinamica di $hTaff_{t+1}$

Tabella C.5: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a $hTaff_{t+1}$.

Passo 0

Variabile di output	$hTaff_{t+1}$
Varianza variabile di output	$4.6282 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	28.2354	$1.3068 \cdot 10^6$	0.4992
s_t	10.3975	481213	0.3541
h_t	9.3068	430736	0.3547
$t_{mod\ T}$	8.3028	384267	0.4188
$gateTaff_t$	7.2724	336581	0.3210

Passo 1

Variabile di output	$hTaff_{t+1} - \hat{r}^0_{t+1}$
Varianza variabile di output	$2.3173 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$t_{mod\ T}$	8.9761	208002	0.0954
s_t	5.8530	135630	0.0315
h_t	4.7180	109328	0.0327
$Air - Temp_t$	4.6113	106857	0.2456
T^{sed}_t	3.1516	73030.5	0.0313

Passo 1.1

Variabile di output	$hTaff_{t+1} - \hat{r}^1_{t+1}$
Varianza variabile di output	$1.9779 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$t_{mod\,T}$	9.7537	192916	0.0790
$Vap - Press_t$	2.2673	44844.5	0.2150
h_t	2.2574	44647.8	0.0070
$Cloud - Cover_t$	2.2219	43946.3	0.1300
T_t^{sed}	2.1867	43249.9	0.0254

Passo 1.2

Variabile di output	$hTaff_{t+1} - \hat{r}^2_{t+1}$
Varianza variabile di output	$1.7627 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$t_{mod\,T}$	8.5329	150406	0.0654
h_t	2.2440	39553.3	0.0036
s_t	2.2102	38958.4	0.0028
T_t^{sed}	2.0333	35840.8	0.0176
T_t^{bot}	1.8028	31776.5	0.0160

Passo 1.3

Variabile di output	$hTaff_{t+1} - \hat{r}^3_{t+1}$
Varianza variabile di output	$1.3307 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.65385	35314	-0.0100
s_t	2.61453	34790.8	-0.0098
$Cloud - Cover_t$	1.68263	22390.3	0.0903
u_t^{-13}	1.57854	21005.2	-0.0013
T_t^{-7}	1.42802	19002.2	0.0092

Passo 1.4

Variabile di output	$hTaff_{t+1} - \hat{r}^4_{t+1}$
Varianza variabile di output	$1.2634 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	2.8337	35801.2	-0.0099
h_t	2.4504	30958.5	-0.0103
u_t^{-13}	1.8895	23871.6	-0.0014
T_t^{bot}	1.3497	17051.5	0.0025
T_t^{-7}	1.3106	16558.6	0.0063

Passo 1.5

Variabile di output	$hTaff_{t+1} - \hat{r}^5_{t+1}$
Varianza variabile di output	$1.1577 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	2.7316	31623.4	-0.0095
h_t	2.5206	29181.7	-0.0097
u_t^{-13}	2.3012	26640.9	-0.0016
$gateTaff_t$	1.2107	14016.6	-0.0028
u_t^{-3}	0.9009	10429.6	-0.0034

C.5.2 Dinamica di T_{t+1}^{sed}

Tabella C.6: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a T_{t+1}^{sed}.

Passo 0

Variabile di output	T_{t+1}^{sed}
Varianza variabile di output	924057

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{sed}	52.3623	483857	0.8078
$t_{mod\ T}$	10.0115	92512.2	0.6364
T_t^{-13}	5.2037	48085.2	0.5123
T_t^{bot}	4.6152	42647.2	0.5434
$Vap - Press_t$	3.0566	28244.6	0.5473

Passo 1

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{0}$
Varianza variabile di output	173041

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	5.0226	8691.12	0.0006
u_t^{-13}	4.9915	8637.35	0.0011
$hTaff_t$	4.9713	8602.41	-0.0067
T_t^{-13}	4.2740	7395.88	0.0081
T_t^{-7}	4.1429	7168.95	0.0091

Passo 1.1

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{1}$
Varianza variabile di output	126783

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	5.5774	7071.19	-0.0003
u_t^{-13}	5.5369	7019.82	-0.0005
$hTaff_t$	4.3390	5501.17	-0.0014
h_t	2.7796	3524.09	-0.0077
T_t^{-3}	2.6623	3375.31	-0.0156

Passo 1.2

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{2}$
Varianza variabile di output	120703

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	5.0008	6036.12	9.0635e-05
$hTaff_t$	4.6097	5564.01	0.0076
T_t^{-3}	2.4162	2916.39	-0.0062
$t_{mod\ T}$	2.3943	2889.99	0.0077
T_t^{-13}	2.3740	2865.47	-0.0076

Passo 1.3

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{3}$
Varianza variabile di output	98296.4

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	4.2979	4224.64	-0.0027
u_t^{-13}	3.6613	3598.91	0.0011
h_t	2.3409	2301.01	-0.0053
s_t	1.8080	1777.24	-0.0112
T_t^{-13}	1.6030	1575.71	-0.0088

Passo 1.4

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{4}$
Varianza variabile di output	89901.9

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	4.2001	3775.97	0.0010
h_t	1.8498	1663.01	-0.0003
s_t	1.5038	1351.91	-0.0038
T_t^{-13}	1.3590	1221.75	-0.0037
T_t^{-3}	1.2974	1166.43	-0.0028

Passo 1.5

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{5}$
Varianza variabile di output	84045.1

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6548	1390.82	-0.0023
s_t	1.3137	1104.06	-0.0078
T_t^{-13}	1.2327	1036.02	-0.0037
T_t^{-3}	1.1719	984.92	-0.0031
$Rain_t$	1.1451	962.41	0.0029

Passo 1.6

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{6}$
Varianza variabile di output	81447.7

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6789	1367.44	-0.0031
s_t	1.3333	1085.99	-0.0091
T_t^{-13}	1.1615	946.03	-0.0043
T_t^{-3}	1.1448	932.39	-0.0033
DO_t^F	0.8117	661.09	0.0130

Passo 1.7

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{7}$
Varianza variabile di output	80461.2

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.7600	1416.12	-0.0037
s_t	1.3753	1106.62	-0.0102
T_t^{-13}	1.1491	924.60	-0.0033
T_t^{-3}	1.1017	886.46	-0.0029
$gateTaff_t$	0.8310	668.66	-0.0010

Passo 1.8

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{8}$
Varianza variabile di output	79687.5

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6064	1280.11	-0.0029
s_t	1.3336	1062.71	-0.0098
T_t^{-13}	1.0996	876.22	-0.0028
T_t^{-3}	1.0690	851.90	-0.0025
DO_t^K	0.7394	589.22	0.0118

Passo 1.9

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{9}$
Varianza variabile di output	79535.4

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6394	1303.92	-0.0021
s_t	1.3750	1093.61	-0.0085
T_t^{-13}	1.1040	878.04	-0.0035
T_t^{-3}	1.0608	843.75	-0.0029
T_t^F	0.7262	577.59	0.0073

Passo 1.10

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{10}$
Varianza variabile di output	78861.5

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6644	1312.58	-0.0041
s_t	1.3598	1072.40	-0.0112
T_t^{-13}	1.0891	858.89	-0.0038
T_t^{-3}	1.0804	851.98	-0.0033
T_t^{bot}	0.7261	572.63	-0.0041

Passo 1.11

Variabile di output	$T_{t+1}^{sed} - \hat{r}_{t+1}^{9}$
Varianza variabile di output	77840.7

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	1.6523	1286.16	-0.0035
s_t	1.3502	1050.98	-0.0105
T_t^{-13}	1.0321	803.41	-0.0035
T_t^{bot}	0.7083	551.31	-0.0043
$Vap - Press_t$	0.6944	540.54	0.0084

C.5.3 Dinamica di h_{t+1}

Tabella C.7: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a h_{t+1}.

Passo 0

Variabile di output	h_{t+1}
Varianza variabile di output	$2.7534 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	49.0989	$1.3519 \cdot 10^6$	0.9815
h_t	48.4466	$1.3339 \cdot 10^6$	0.9820
u_t^{-13}	0.3621	9969.99	-0.0025
u_t^{-3}	< 0.3	n/a	-0.0012
$hTaff_t$	< 0.3	n/a	0.4256

Passo 1

Variabile di output	$h_{t+1} - \hat{r}_{t+1}^0$
Varianza variabile di output	49277.9

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	20.4920	10098	-0.0851
u_t^{-3}	19.6318	9674.12	-0.0948
s_t	8.3099	4094.96	-0.4302
$SSOL4_t^K$	1.5644	770.924	-0.0103
$SSOL3_t^F$	1.5354	756.612	-0.0127

Passo 1.1

Variabile di output	$h_{t+1} - \hat{r}^1_{t+1}$
Varianza variabile di output	28759.6

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	33.5756	9656.23	-0.1441
u_t^{-3}	33.3308	9585.83	-0.1541
s_t	7.6294	2194.18	-0.8584
$Rain_t$	0.5506	158.358	-0.7769
$gateTSSmax_t$	0.5211	149.87	-0.8023

Passo 1.2

Variabile di output	$h_{t+1} - \hat{r}^2_{t+1}$
Varianza variabile di output	16652.8

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	48.2385	8033.08	0.2708
s_t	6.8195	1135.64	-0.3388
SW_t	1.0432	173.72	-0.2207
$Rain_t$	1.0113	168.42	-0.2488
$gateTaff_t$	0.6394	106.49	-0.2918

Passo 1.3

Variabile di output	$h_{t+1} - \hat{r}^3_{t+1}$
Varianza variabile di output	6759.19

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	6.7279	454.75	-0.1199
SW_t	3.8889	262.86	0.1880
$Rain_t$	2.4580	166.14	0.0341
$hTaff_t$	0.6843	46.2544	-0.0920
$Vap - Press_t$	0.6349	42.9142	0.1651

Passo 1.4

Variabile di output	$h_{t+1} - \hat{r}^4_{t+1}$
Varianza variabile di output	6730.28

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	7.0979	477.71	-0.1265
$Rain_t$	2.4478	164.75	0.0275
$hTaff_t$	0.6581	44.29	-0.0949
$PO^K_{4,t}$	0.6477	43.59	0.0625
$NO^F_{3,t}$	0.6328	42.59	0.1984

Passo 1.5

Variabile di output	$h_{t+1} - \hat{r}^5_{t+1}$
Varianza variabile di output	6704.33

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	7.2297	484.70	-0.1434
$Rain_t$	2.6178	175.51	0.0097
$hTaff_t$	0.8161	54.72	-0.1143
$Vap - Press_t$	0.5912	39.63	0.1282
$SSOL5^F_t$	0.5586	37.45	0.1721

Passo 1.6

Variabile di output	$h_{t+1} - \hat{r}^6_{t+1}$
Varianza variabile di output	6613.13

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	7.4317	491.47	-0.1545
$Rain_t$	2.4280	160.57	-0.0087
$hTaff_t$	0.7937	52.49	-0.1217
$Vap - Press_t$	0.5255	34.75	0.1114
$SSOL6^K_t$	0.4868	32.20	0.1531

Passo 1.7

Variabile di output	$h_{t+1} - \hat{r}^7_{t+1}$
Varianza variabile di output	6595.16

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	7.6176	502.39	-0.1555
$Rain_t$	2.2835	150.60	-0.0157
$hTaff_t$	0.7222	47.63	-0.1251
$SSOL6^F_t$	0.5069	33.43	0.1443
$SSOL3^F_t$	0.4935	32.55	0.1213

Passo 1.8

Variabile di output	$h_{t+1} - \hat{r}^8_{t+1}$
Varianza variabile di output	6562.43

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	7.2210	473.87	-0.1595
$Rain_t$	2.1913	143.80	-0.0159
$hTaff_t$	0.7796	51.16	-0.1288
$gateTSSmax_t$	0.5089	33.40	-0.1155
$SSOL2^F_t$	0.5055	33.17	0.0949

C.5.4 Dinamica di T_{t+1}^{-7}

Tabella C.8: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a T_{t+1}^{-7}.

Passo 0

Variabile di output	T_{t+1}^{-7}
Varianza variabile di output	$1.4442 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^K	15.7985	228155	0.7728
T_t^{-3}	13.7239	198195	0.8118
T_t^F	13.1026	189222	0.7710
DO_t^K	10.6259	153454	0.7614
T_t^{-7}	10.2841	148518	0.8104

Passo 1

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^0$
Varianza variabile di output	265114

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	10.7430	28481.3	0.0071
u_t^{-13}	8.2200	21792.4	0.0011
$Vap - Press_t$	3.3249	8814.67	0.0097
h_t	2.9334	7776.95	-0.0243
$hTaff_t$	2.7456	7278.98	-0.0168

Passo 1.1

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{1}$
Varianza variabile di output	236758

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	11.6356	27548.1	0.0090
u_t^{-13}	9.6927	22948.3	0.0018
h_t	3.8634	9146.83	-0.0315
$hTaff_t$	3.3246	7871.21	-0.0234
s_t	2.8597	6770.5	-0.0312

Passo 1.2

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{2}$
Varianza variabile di output	214223

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	10.0578	21546.1	0.0085
$hTaff_t$	3.9235	8405.09	-0.0034
h_t	3.2369	6934.21	-0.0121
s_t	2.2994	4926.03	-0.0115
T_t^{bot}	2.2548	4830.22	-0.0105

Passo 1.3

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{3}$
Varianza variabile di output	193660

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	3.4969	6772.01	0.0024
h_t	2.9949	5799.91	-0.0049
T_t^{-13}	2.9181	5651.14	0.0037
T_t^{-7}	2.7115	5251.13	0.0008
T_t^{bot}	2.6374	5107.61	-0.0042

Passo 1.4

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{4}$
Varianza variabile di output	173737

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	3.25765	5659.74	0.0001
T_t^{bot}	2.53478	4403.85	-0.0012
T_t^{-7}	2.52075	4379.48	-0.0029
$hTaff_t$	2.47011	4291.5	0.0012
s_t	2.37726	4130.18	0.0002

Passo 1.5

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{5}$
Varianza variabile di output	168153

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	2.2441	3773.52	-0.0028
T_t^{-7}	2.1802	3666.16	-0.0031
T_t^{bot}	2.0458	3440.13	-0.0030
s_t	1.9633	3301.42	-0.0023
$NH_{4,t}^{F}$	1.5518	2609.41	0.0308

Passo 1.6

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{6}$
Varianza variabile di output	153978

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
h_t	4.0048	6166.51	-0.0024
T_t^{bot}	3.2022	4930.66	-0.0052
s_t	3.1422	4838.26	-0.0052
T_t^{-7}	2.7585	4247.48	-0.0033
T_t^{sed}	1.0740	1653.74	-0.0045

Passo 1.7

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{7}$
Varianza variabile di output	136667

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.0866	2851.73	-0.0055
T_t^{-7}	1.8201	2487.53	-0.0043
$Cloud - Cover_t$	1.4146	1933.27	0.0057
SW_t	1.2826	1752.83	0.0169
T_t^K	1.1478	1568.62	0.0183

Passo 1.8

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{8}$
Varianza variabile di output	131918

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.1012	2771.85	-0.0068
T_t^{-7}	1.9379	2556.51	-0.0045
SW_t	0.9348	1233.21	0.0130
$PO_{4,t}^F$	0.9167	1209.37	0.0090
s_t	0.8999	1187.23	-0.0072

Passo 1.9

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{9}$
Varianza variabile di output	131470

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.1981	2889.81	-0.0071
T_t^{-7}	2.0370	2678.04	-0.0048
s_t	1.1440	1504.01	-0.0075
NH_t^K	1.0500	1380.4	0.0152
T_t^{sed}	0.9131	1200.49	-0.0022

Passo 1.10

Variabile di output	$T_{t+1}^{-7} - \hat{r}_{t+1}^{10}$
Varianza variabile di output	130424

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{bot}	2.3539	3070.03	-0.0065
T_t^{-7}	2.1227	2768.49	-0.0039
s_t	1.3125	1711.8	-0.0080
$Cloud - Cover_t$	1.0810	1409.89	0.0039
T_t^{sed}	0.9563	1247.22	-0.0024

C.5.5 Dinamica di TSS_{t+1}^{-3}

Tabella C.9: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a TSS_{t+1}^{-3}.

Passo 0

Variabile di output	TSS_{t+1}^{-3}
Varianza variabile di output	$1.0597 \cdot 10^8$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$Rain_t$	8.4622	$8.9672 \cdot 10^6$	0.6137
$SSOL3_t^F$	6.7472	$7.1499 \cdot 10^6$	0.7688
$SSOL4_t^K$	6.1430	$6.5096 \cdot 10^6$	0.7636
$SSOL2_t^K$	5.8496	$6.1986 \cdot 10^6$	0.7593
$SSOL1_t^F$	5.8456	$6.1943 \cdot 10^6$	0.7703

Passo 1

Variabile di output	$TSS_{t+1}^{-3} - \hat{r}_{t+1}^0$
Varianza variabile di output	$1.9415 \cdot 10^7$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
TSS_t^{-3}	10.2124	$1.9827 \cdot 10^6$	0.1776
TSS_t^{-7}	5.7063	$1.1079 \cdot 10^6$	0.1242
TSS_t^{bot}	2.1366	414828	0.0657
$DOCL_t^K$	2.0147	391158	0.0951
$DOCL_t^F$	1.7563	340998	0.0979

Passo 1.1

Variabile di output	$TSS_{t+1}^{-3} - \hat{r}_{t+1}^{1}$
Varianza variabile di output	$1.5886 \cdot 10^7$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$DOCL_t^F$	1.4811	235296	0.0675
$DOCL_t^K$	1.4250	226369	0.0708
$SSOL6_t^K$	1.2578	199817	0.0861
$SSOL2_t^F$	1.2074	191806	0.0938
$SSOL1_t^K$	1.1244	178626	0.0727

Passo 1.2

Variabile di output	$TSS_{t+1}^{-3} - \hat{r}_{t+1}^{2}$
Varianza variabile di output	$1.5273 \cdot 10^7$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.1334	325838	0.0026
$DOCL_t^K$	1.8003	274961	0.0489
$DOCL_t^F$	1.6356	249801	0.0481
$POCL_t^K$	1.0680	163117	0.0802
TSS_t^{bot}	1.0634	162408	-0.0099

Passo 1.3

Variabile di output	$TSS_{t+1}^{-3} - \hat{r}_{t+1}^{3}$
Varianza variabile di output	$1.4839 \cdot 10^7$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.3534	349217	0.0019
$DOCL_t^F$	2.1484	318798	0.0325
$DOCL_t^K$	1.7523	260034	0.0341
TSS_t^{bot}	1.1317	167931	-0.0088
u_t^{-3}	0.9420	139790	-0.0015

Passo 1.4

Variabile di output	$TSS_{t+1}^{-3} - \hat{r}_{t+1}^{4}$
Varianza variabile di output	$1.43496 \cdot 10^7$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	1.8499	265467	0.0013
TSS_t^{bot}	0.9048	129829	-0.0118
$DOCL_t^F$	0.8209	117806	0.0195
$SSOL3_t^K$	0.7556	108423	0.0170
$SSOL6_t^F$	0.7055	101233	0.0186

C.5.6 Dinamica di T_{t+1}^{-3}

Tabella C.10: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a T_{t+1}^{-3}.

Passo 0

Variabile di output	T_{t+1}^{-3}
Varianza variabile di output	$1.6091 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^K	18.5253	298096	0.7811
T_t^{-3}	15.1692	244093	0.8253
T_t^F	13.1117	210985	0.7807
T_t^{-7}	10.4639	168379	0.8050
DO_t^K	10.1239	162906	0.7694

Passo 1

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^0$
Varianza variabile di output	274018

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-3}	10.2335	28041.6	0.0067
u_t^{-13}	9.5470	26160.5	0.0015
h_t	7.0409	19293.4	-0.0052
s_t	5.6663	15526.7	-0.0058
T_t^{-13}	3.3166	9088.17	-0.0289

Passo 1.1

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{1}$
Varianza variabile di output	257023

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	9.0086	23154.2	0.0065
h_t	7.0714	18175.2	0.0182
s_t	6.4498	16577.4	0.0169
T_t^{-13}	4.0684	10456.7	0.0016
T_t^{-7}	3.5786	9197.7	-0.0047

Passo 1.2

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{2}$
Varianza variabile di output	203808

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	7.5553	15398.4	0.0077
Vap_Press_t	3.1669	6454.51	0.0336
T_t^{-7}	2.6871	5476.46	-0.0122
Air_Temp_t	2.6635	5428.44	0.0381
$hTaff_t$	2.5063	5108.13	0.0038

Passo 1.3

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{3}$
Varianza variabile di output	169913

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	9.6965	16475.5	0.0099
$hTaff_t$	3.0912	5252.28	-0.0083
s_t	2.5662	4360.27	-0.0176
T_t^F	1.7570	2985.36	0.0083
T_t^K	1.6148	2743.71	0.0093

Passo 1.4

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{4}$
Varianza variabile di output	140117

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.4929	3492.94	-0.0008
$NH_{4,t}^{F}$	1.7485	2449.89	0.0378
$NH_{4,t}^{K}$	1.6801	2354.16	0.0378
T_t^{bot}	1.6775	2350.49	-0.0051
T_t^{-7}	1.6037	2247.07	-0.0025

Passo 1.5

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{5}$
Varianza variabile di output	130311

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.6423	3443.22	-0.0009
T_t^{bot}	2.0510	2672.74	-0.0060
T_t^{-13}	1.6796	2188.74	-0.0030
T_t^{-7}	1.5758	2053.46	-0.0029
$Rain_t$	1.3064	1702.38	0.0011

Passo 1.6

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{6}$
Varianza variabile di output	128758

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.4720	3182.88	-0.0012
T_t^{bot}	2.2094	2844.79	-0.0062
T_t^{-13}	1.8113	2332.17	-0.0034
T_t^{-7}	1.6365	2107.11	-0.0034
T_t^{K}	1.1484	1478.7	0.0151

Passo 1.7

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{7}$
Varianza variabile di output	125790

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff$	2.8457	3579.63	-0.0011
T_t^{bot}	2.0297	2553.21	-0.0065
T_t^{-13}	1.6438	2067.7	-0.0031
SW_t	1.5431	1941.01	0.0115
T_t^{-7}	1.4933	1878.46	-0.0034

Passo 1.8

Variabile di output	$T_{t+1}^{-3} - \hat{r}_{t+1}^{8}$
Varianza variabile di output	125280

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.6782	3355.29	-0.0013
T_t^{bot}	2.0376	2552.72	-0.0066
T_t^{-13}	1.6810	2105.95	-0.0034
T_t^{-7}	1.6158	2024.29	-0.0039
$Cloud - Cover_t$	1.1538	1445.44	0.0042

C.5.7 Dinamica di T_{t+1}^{-13}

Tabella C.11: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a T_{t+1}^{-13}.

Passo 0

Variabile di output	T_{t+1}^{-13}
Varianza variabile di output	$1.3723 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-13}	16.5787	227513	0.7917
T_t^{-7}	11.8423	162514	0.7852
T_t^{F}	11.2896	154930	0.7428
T_t^{K}	11.2391	154237	0.7467
T_t^{-3}	9.0061	123593	0.7697

Passo 1

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{0}$
Varianza variabile di output	275869

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	7.3705	20333.1	-0.0002
u_t^{-3}	6.3959	17644.2	0.0036
h_t	4.5235	12478.9	-0.0193
T_t^{-7}	4.0325	11124.4	-0.0430
T_t^{-3}	3.7953	10469.9	-0.0363

Passo 1.1

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{1}$
Varianza variabile di output	256807

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
u_t^{-13}	6.8619	17621.7	0.0029
T_t^{-7}	3.6416	9351.86	-0.0146
T_t^{-3}	3.5396	9089.83	-0.0092
h_t	3.3805	8681.27	0.0014
s_t	2.7081	6954.63	-0.0039

Passo 1.2

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{2}$
Varianza variabile di output	239179

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	3.5805	8563.84	-0.0071
T_t^{-3}	3.5786	8559.32	-0.0036
h_t	2.9227	6990.55	0.0108
T_t^{bot}	2.7069	6474.38	-0.0071
$hTaff_t$	2.5519	6103.71	0.0117

Passo 1.3

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{3}$
Varianza variabile di output	209377

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	3.1052	6501.63	-0.0069
T_t^{-3}	2.9409	6157.61	-0.0060
T_t^{bot}	2.6473	5542.77	-0.0066
h_t	2.3295	4877.49	0.0032
s_t	1.8763	3928.63	0.0026

Passo 1.4

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{4}$
Varianza variabile di output	177553

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-3}	2.8910	5133.1	-0.0061
T_t^{-7}	2.7680	4914.74	-0.0056
T_t^{bot}	2.6361	4680.43	-0.0042
$gateTSSmax_t$	1.5888	2820.99	-0.0008
$Vap - Press_t$	1.2162	2159.41	0.0376

Passo 1.5

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{5}$
Varianza variabile di output	158236

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	2.5744	4073.66	-0.0058
T_t^{-3}	2.4376	3857.1	-0.0043
T_t^{bot}	1.8712	2960.97	-0.0060
s_t	1.1221	1775.61	0.0021
T_t^{K}	0.9278	1468.2	0.0209

Passo 1.6

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{6}$
Varianza variabile di output	153957

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	2.4322	3744.49	-0.0044
T_t^{-3}	2.3345	3594.16	-0.0035
T_t^{bot}	1.6471	2535.78	-0.0069
s_t	1.1444	1761.83	0.0023
$Rain_t$	1.0385	1598.92	0.0034

Passo 1.7

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{7}$
Varianza variabile di output	151798

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	2.4470	3714.48	-0.0043
T_t^{-3}	2.3498	3567.02	-0.0035
T_t^{bot}	1.6072	2439.64	-0.0077
s_t	1.1641	1767.12	0.0023
DO_t^K	0.8867	1345.95	0.0070

Passo 1.8

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{8}$
Varianza variabile di output	150937

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	2.4505	3698.72	-0.0044
T_t^{-3}	2.3373	3527.85	-0.0037
T_t^{bot}	1.6359	2469.19	-0.0080
s_t	1.1680	1762.96	0.0018
T_t^{F}	0.8841	1334.49	0.0055

Passo 1.9

Variabile di output	$T_{t+1}^{-13} - \hat{r}_{t+1}^{9}$
Varianza variabile di output	150661

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
T_t^{-7}	2.4428	3680.38	-0.0043
T_t^{-3}	2.1855	3292.71	-0.0035
T_t^{bot}	1.6127	2429.76	-0.0081
s_t	1.1706	1763.71	0.0019
DO_t^K	0.8031	1209.9	0.0059

C.5.8 Dinamica di $gateTaff_{t+1}$

Tabella C.12: Risultati ottenuti tramite l'algoritmo IFR nel selezionare le variabili più rappresentative rispetto a $gateTaff_{t+1}$.

Passo 0

Variabile di output	$gateTaff_{t+1}$
Varianza variabile di output	$4.6990 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$gateTaff_t$	27.1551	$1.2760 \cdot 10^6$	0.3814
h_t	7.6020	357222	0.2922
s_t	7.4223	348777	0.2917
$hTaff_t$	4.4145	207438	0.3134
$Air - Temp_t$	2.8064	131876	0.2788

Passo 1

Variabile di output	$gateTaff_{t+1} - \hat{r}^0_{t+1}$
Varianza variabile di output	$2.9042 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
s_t	4.0031	116257	0.0511
$Air - Temp_t$	3.9833	115682	0.1449
h_t	3.3037	95943.8	0.0518
$hTaff_t$	2.9816	86591.8	0.0228
$t_{mod\ T}$	2.5942	75339.2	0.0679

Passo 1.1

Variabile di output	$gateTaff_{t+1} - \hat{r}^1_{t+1}$
Varianza variabile di output	$2.3708 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	3.3328	79014.2	0.0117
s_t	2.6133	61957	0.0218
h_t	2.4181	57328.8	0.0239
$t_{mod\ T}$	1.8484	43821.2	0.0294
$gateTSSmax_t$	1.8106	42925.5	0.0029

Passo 1.2

Variabile di output	$gateTaff_{t+1} - \hat{r}^2_{t+1}$
Varianza variabile di output	$2.0643 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	3.4881	72006.2	0.0060
$gateTSSmax_t$	1.9615	40491.9	0.0012
s_t	1.8273	37722	0.0038
h_t	1.6971	35034.1	0.0064
T_t^{bot}	1.3156	27157.7	-0.0027

Passo 1.3

Variabile di output	$gateTaff_{t+1} - \hat{r}^3_{t+1}$
Varianza variabile di output	$1.9038 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	2.8784	54798.1	0.0011
T_t^{bot}	1.2667	24115.1	-0.0043
$Rain_t$	1.1948	22745.6	0.0018
s_t	1.0298	19605.3	-0.0040
$Cloud - Cover_t$	0.9971	18982.8	0.0267

Passo 1.4

Variabile di output	$gateTaff_{t+1} - \hat{r}^4_{t+1}$
Varianza variabile di output	$1.8426 \cdot 10^6$

Feature candidata	Punteggio %	Riduzione varianza	Performance SISO (R^2)
$hTaff_t$	3.0318	55864.6	0.0015
T_t^{bot}	1.2625	23263.8	-0.0040
u_t^{-13}	1.1686	21534	-0.0004
s_t	1.0257	18900.1	-0.0042
$Rain_t$	1.0196	18787.4	0.0009

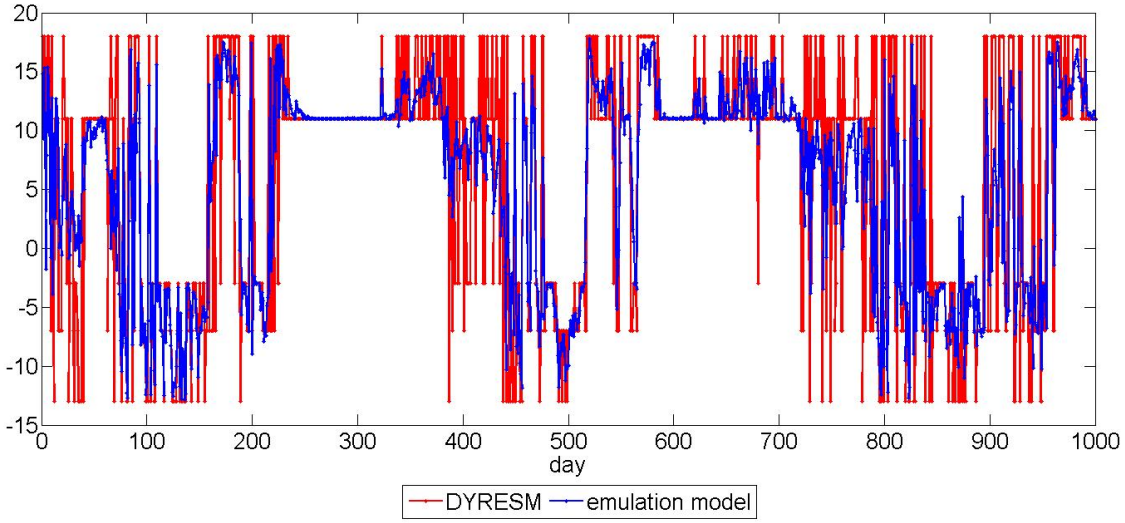

Figura C.1: Traiettorie di $gateTaff_{t+1}$ simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

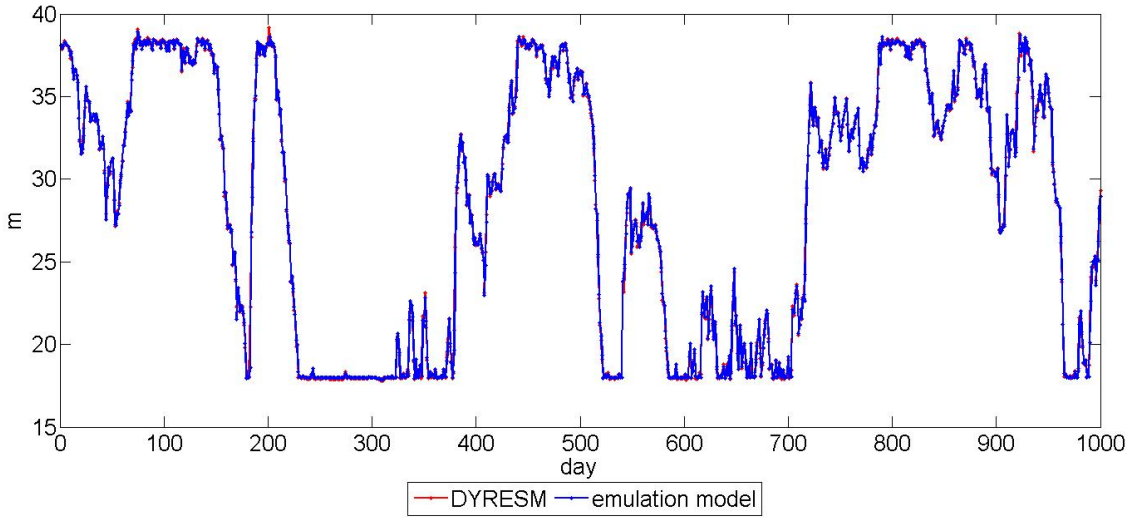

Figura C.2: Traiettorie di h_{t+1} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

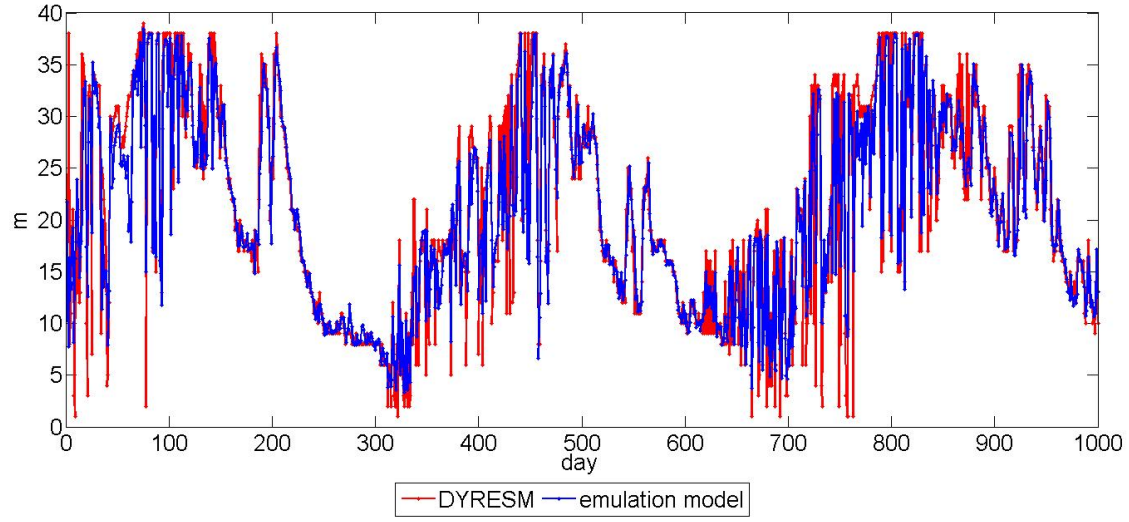

Figura C.3: Traiettorie di $hTaff_{t+1}$ simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

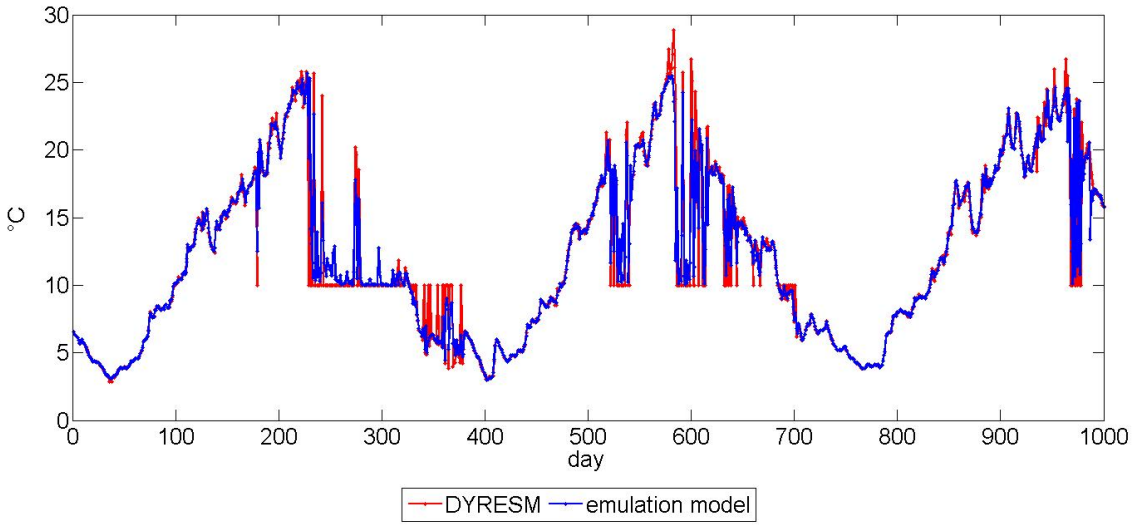

Figura C.4: Traiettorie di T_{t+1}^{-3} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

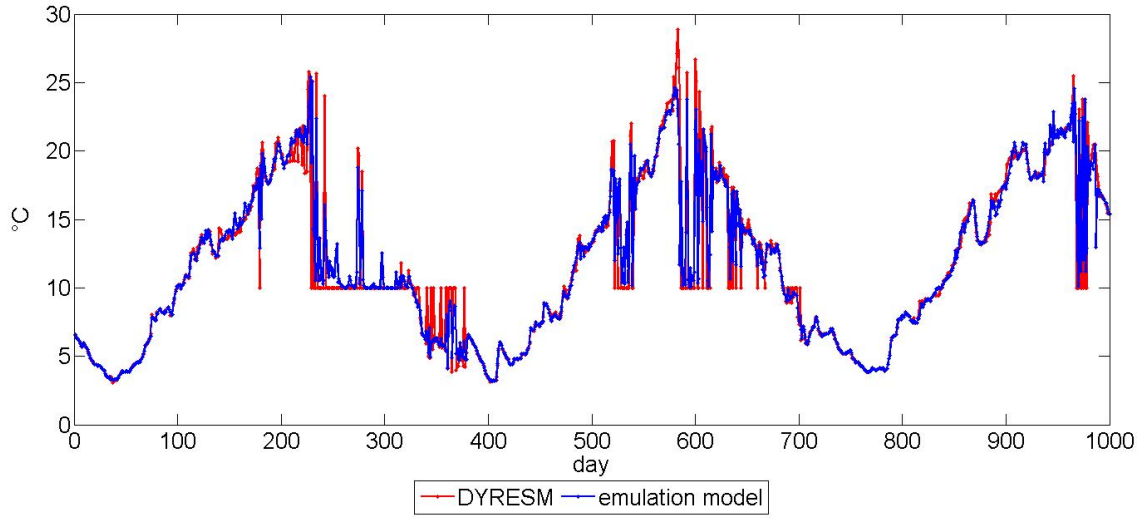

Figura C.5: Traiettorie di T_{t+1}^{-7} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

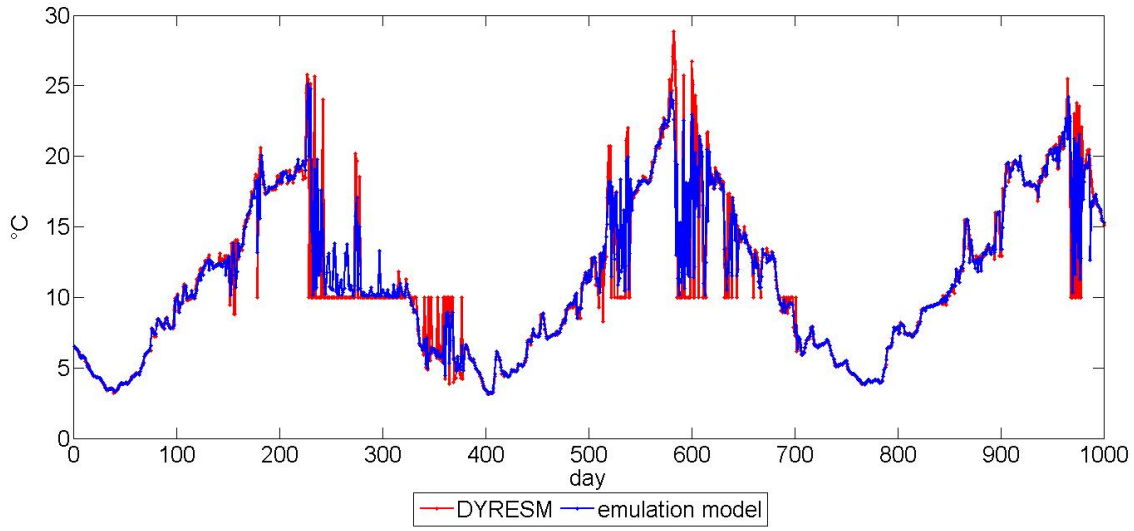

Figura C.6: Traiettorie di T_{t+1}^{-13} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

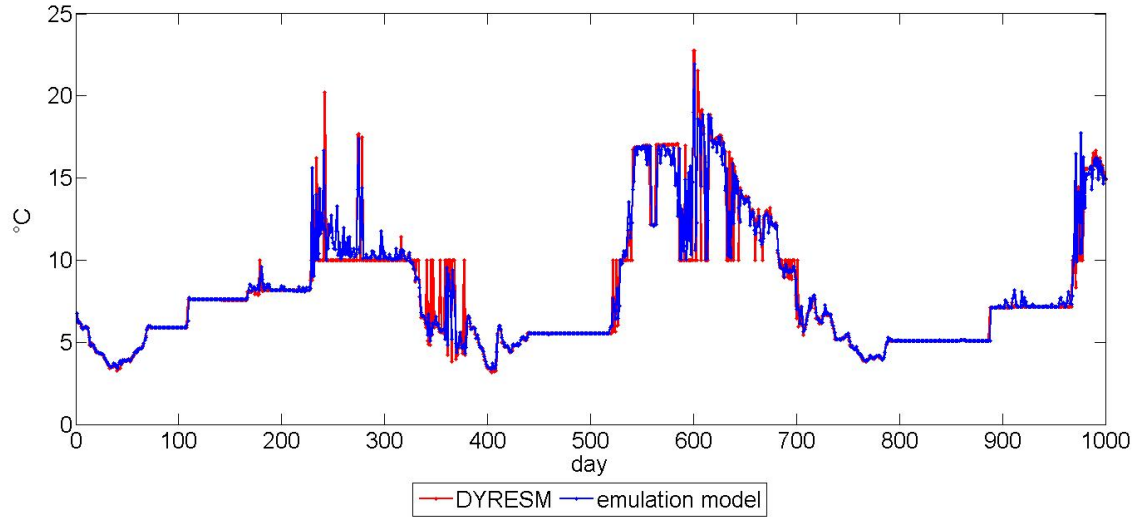

Figura C.7: Traiettorie di T_{t+1}^{sed} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

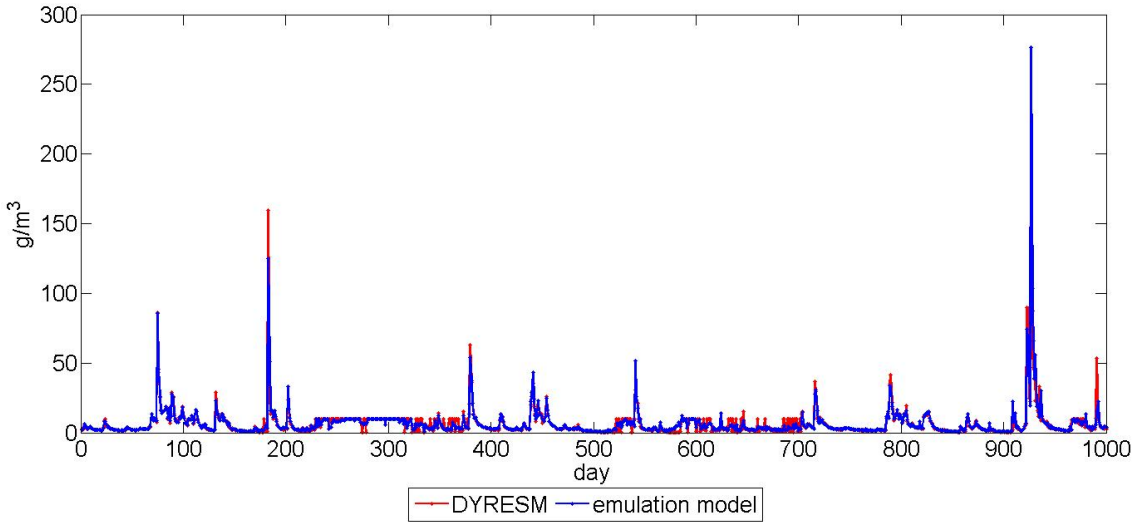

Figura C.8: Traiettorie di TSS_{t+1}^{-3} simulate da DYRESM-CAEDYM (rosso) e dall'emulation model (blu) (periodo 1995-1997).

Bibliografia

[1] A.C. Antoulas, D.C. Sorensen, and S. Gugercin. A survey of model reduction methods for large-scale systems. *Contemporary mathematics*, 280:193–219, 2001.

[2] R. Barto and R. Sutton. *Reinforcement Learning: An Introduction*. MIT Press, Boston, Mass., 1998.

[3] M.B. Beck. Water quality modelling: a review of the analysis of uncertainty. *Water Resources Research*, 23 (8):1393–1442, 1987.

[4] R. Bellman. *Dynamic Programming*. Princeton Univ. Press, Princeton, N.J., 1957.

[5] R. Bellman and S. Dreyfus. *Applied Dynamic Programming*. Princeton Univ. Press, Princeton, N.J., 1962.

[6] R. Bellman, R.Kabala, and B. Kotkin. Polynomial approximation – a newcomputational technique in dynamic programming. *Math. Comupt.*, 17:155–161, 1963.

[7] A. Bennett. *Inverse modeling of the ocean and atmosphere*. Cambridge University Press, Cambridge, U.K., 2002.

[8] K. Bernhardt. Finding alternatives and reduced formulations for process-based models. *Evolutionary computation*, 16(1):63–88, 2008.

[9] D. Bertsekas and J. Tsitsiklis. *Neuro-Dynamic Programming*. Athena Sci., Boston, Mass., 1996.

[10] K. Beven. Changing ideas in hydrology - the case of physically-based models. *Journal of Hydrology*, 105 (1-2):157–172, 1989.

[11] K. Beven. Prophecy, reality and uncertainty in distributed hydrological modelling. *Advances in Water Resources*, 16:41–51, 1993.

[12] K. Beven. A manifesto for the equifinality thesis. *Journal of Hydrology*, 320:18–36, 2006.

[13] C.M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Oxford, U.K., 2004.

[14] R.W. Blanning. The construction and implementation of metamodels. *Simulation*, 24(6):177–184, 1975.

[15] A.F. Blumberg and G.L. Mellor. *Three-Dimensional Coastal Ocean Models*, chapter A description of a three-dimensional coastal ocean circulation model. American Geophysical Union, Washington, D.C., 1987.

[16] G. Box and G. Jenkins. *Time Series Analysis: Forecasting and Control*. Holden-Day Inc., San Francisco, CA, 1970.

[17] G.E.P. Box and K.B. Wilson. On the experimental attainment of optimum conditions (with discussion). *Journal of the Royal Statistical Society Series B*, 13 (1):1–45, 1951.

[18] L. Breiman, J. Friedman, R. Olsen, and C. Stone. *Classification and regression trees*. Wadsworth International, 1984.

[19] A. Castelletti, G. Corani, A. Rizzoli, R. Soncini-Sessa, and E. Weber. A reinforcement learning approach for the operational management of a water system,. In *Modelling and Control in Environmental Issues, Int. Fed. of Autom. Control*, Yokohama (Japan), 22 - 23 August 2001.

[20] A. Castelletti, D. de Rigo, A. Rizzoli, R. Soncini-Sessa, and E. Weber. An improved technique for neuro-dynamic programming applied to the efficient and integrated water resources management,. In *16th World Congress, Int. Fed. of Autom. Control*, Prague, 4 - 8 July 2005.

[21] A. Castelletti, S. Galelli, and R. Soncini-Sessa. A tree-based feature ranking approach to enhance emulation modelling of 3D hydrodynamic-ecological models. In *2010 International Congress on Environmental Modelling and Software*, Ottawa (Canada), 5 - 8 July 2010c.

[22] A. Castelletti, A. Gall, G. Garbarini, R. Soncini-Sessa, and E. Weber. Tono dam optimization. 28 February 2010b.

[23] A. Castelletti, F. Pianosi, and R. Soncini-Sessa. Water reservoir control under economic, social and environmental constraints. *Automatica*, 44:1595–1607, 2008.

[24] C. Cervellera, V. Chen, and A. Wen. Optimization of a large-scale water reservoir network by stochastic dynamic programming with efficient space discretization. *Eur.J. Oper. Res.*, 171:1139–1151, 2006.

[25] D. Cohn, Z. Ghahramani, and M. Jordan. Active learning with statistical models. *J. Artificial Intelligence Res.*, 4:129–145, 1996.

[26] R. Courant, K. Friedrichs, and H. Lewy. On the partial difference equations of mathematical physics. *IBM Journal*, 11:215–234, 1967.

[27] G.M. Cox, J.M. Gibbons, A.T.A. Wood, J. Craigon, S.J. Ramsden, and N.M.J. Crout. Towards the systematic simplification of mechanistic models. *Ecological modelling*, 198:240–246, 2006.

[28] N.M.J. Crout, D. Tarsitano, and A.T. Wood. Is my model too complex? evaluating model formulation using model reduction. *Environmental modelling & software*, 24:1–7, 2009.

[29] D. Ernst, P. Geurts, and L. Wehenkel. Tree-based batch mode reinforcement learning. *J. Mach. Learn. Res.*, 6:503–556, 2005.

[30] S. Galelli. *Dealing with complexity and dimensionality in water resources management*. PhD thesis, Politecnico di Milano – Dipartimento di Elettronica e Informazione, 2010. Submitted.

[31] S. Galelli, C. Gandolfi, R. Soncini-Sessa, and D. Agostani. Building a metamodel of an irrigation district distributed-parameter mode. *Agricultural Water Management*, 97:187–200, 2010.

[32] S. Galelli, C. Gandolfi, R. Soncini-Sessa, and D. Agostani. Building a metamodel of an irrigation district distributed-parameter model. *Agricultural Water Management*, 97(2):187–200, 2010.

[33] S. Galelli, F. Pianosi, and R. Soncini-Sessa. Meta-model of an irrigation district distributed parameter model. In *17th IFAC World Congress*, Seoul, K, July 6-11 2008.

[34] S. Galelli, T. Shintani, F. Pianosi, J. Imberger, and R. Soncini-Sessa. Deriving an emulation model of a rectangular-basin two-layer numerical model. In *18th World IMACS/MODSIM Congress*, Cairns (Australia), 13 - 17 July 2009.

[35] A. Galli. Gestione di un serbatoio con sistema a rilascio selettivo per l'integrazione di parametri qualitativi e quantitativi, Politecnico di Milano, A.A. 2008/2009.

[36] G. Garbarini. Gestione integrata di qualità e quantità dell'acqua in un serbatoio con rilascio selettivo, Politecnico di Milano, A.A. 2008/2009.

[37] C. Gaskett. *Q-learning for robot control*. PhD thesis, Australian National University, Camberra, 2002.

[38] I. Guyon and A. Elisseeff. An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3:1157 – 1182, 2003.

[39] M. Hejazi, X. Cai, and B. Ruddel. The role of hydrologic information in reservoir operation – learning from historical releases. *Water Resour.*, 31:1636–1650, 2008.

[40] M.R. Hipsey, J.R. Romero, J.P. Antenucci, and D.P. Hamilton. Computational Aquatic Ecosystem Dynamics Model: Caedym v2.3 User Manual. CWR tech. report, Centre for water Research - University of Western Australia, Crawley - Western Australia, 2006.

[41] J. Imberger and J.C. Patterson. Physical limnology. *Advances in Applied Mechanics*, 27:303–475, 1989.

[42] A. Imerito. Dynamic REservoir Simulation Model: Dyresm Science Manual. CWR tech. report, Centre for water Research - University of Western Australia, Crawley - Western Australia, 2007.

[43] H. Jacobson and Q. Maine. *Differential Dynamic Programming*. Elsevier, New York, 1970.

[44] A.K. Jain, M.N. Murty, and P.J. Flynn. Data clustering: a review. *ACM Computing Surveys*, 31 (3):264–323, 1999.

[45] I.T. Jollife. *Principal component analysis*. Springer-Verlag, New York, N.Y., 1986.

[46] E. Kalnay. *Atmospheric Modeling, Data Assimilation, and Predictability*. Cambridge University Press, Cambridge, U.K., 2002.

[47] J.P.C. Kleijnen. Response surface methodology for constrained simulation optimization: An overview. *Simulation Modelling Practice and Theory*, 16 (1):50–64, 2008.

[48] J.P.C. Kleijnen, S.M. Sanchez, T.W. Lucas, and T.M. Cioppa. A user's guide to the brave new world of designing simulation experiments. *INFORMS Journal of Computing*, 17(3):263–289, 2005.

[49] R. Larson. *State Incremental Dynamic Programming*. Elsevier, New York, 1968.

[50] D.P. Loucks, J.R. Stedinger, and D.A. Haith. *Water resource systems, planning and analysis*. Prentice-Hall International. Inc, Englewood Cliffs, N.J., 1980.

[51] D. Luenberger. Cycling dynamic programming: A procedure for problems with fixed delay. *Oper.Res.*, 19:1101–1110, 1971.

[52] D.C. Montgomery. *Design and Analysis of Experiments, 5th ed.* Wiley, New York, N.Y., 2000.

[53] M.S. Okino and M.L. Mavrovouniotis. Simplification of mathematical models of chemical reaction systems. *Chemical Reviews*, 98 (2):391–408, 1998.

[54] N.V. Queipo, R.T. Haftka, W. Shyy, T. Goel, R. Vaidyanathan, and P.K. Tucker. Surrogate-based analysis and optimization. *Progress in Aerospace Sciences*, 41:1–28, 2005.

[55] M. Ratto, A. Pagano, and P.C. Young. State dependent parameter metamodelling and sensitivity analysis. *Computer Physics communications*, 177(11):863–876, 2007.

[56] A. Saltelli, K. Chan, and M. Scott. *Sensitivity Analysis*. Wiley, New York, USA, 2000.

[57] S.Y. Shvartsman and I.G. Kevrekidis. Nonlinear model reduction for control of distributed systems: a computer-assisted study. *AIChE Journal*, 44 (7):1579–1595, 1998.

[58] T.W. Simpson, J.D. Peplinski, P.N. Koch, and J.K. Allen. Metamodels for computer based engineering design: survey and recommendations. *Engineering with Computers*, 17:129–150, 2001.

[59] R. Soncini-Sessa, A. Castelletti, and E. Weber. *Integrated and participatory water resources management. Theory.* Elsevier, Amsterdam, NL, 2007.

[60] J. Tejada-Guibert, S. Johnson, and J. Stedinger. The value of hydrologic information in stochastic dynamic programming models of a multireservoir system. *Water Resour. Res.*, 31:2571–2579, 1995.

[61] A. Turgeon. A decomposition method for the long-term scheduling of reservoirs in series. *Water Resour. Res.*, 17:1565–1570, 1981.

[62] R. van der Merwe, T.K. Leen, Z. Lu, S. Frolov, and A.M. Baptista. Fast neural network surrogates for very high dimensional physics-based models in computational oceanography. *Neural Networks*, 20:462–478, 2007.

[63] E.H. van Nes and M. Scheffer. A strategy to improve the contribution of complex simulation models to ecological theory. *Ecological modelling*, 185:153–164, 2005.

[64] K. van Werkhoven, T. Wagener, P. Reed, and Y. Tang. Sensitivity-guided reduction of parametric dimensionality for multi-objective calibration of watershed models. *Advances in Water Resources*, 32:1154–1169, 2009.

[65] L. von Bertalanffy. *General Systems Theory.* Braziller, New York, USA, 1968.

[66] C. Watkins and P. Dayan. Q-learning. *Mach. Learn.*, 8:279–292, 1992.

[67] P. Wong and D. Luenberger. Reducing the memory requirements of dynamic programming. *Oper.Res.*, 16:673–696, 1968.

[68] H. Yajima, S. Kikkawa, and J. Ishiguro. Effect of selective withdrawal system operation on the long-and short-term water conservation in a reservoir. *Annual Journal of Hydraulic Engineering*, 50:1375–1380, 2006. (In japanese).

[69] Z. Yang, V.C.P. Chen, M.E. Chang, and T.E. Murphy J.C.C. Tsai. Mining and modeling for a metropolitan atlanta ozone pollution decision-making framework. *IEEE Transactions*, 39:607–615, 2007.

[70] P.C. Young. Data-based mechanistich modelling of environmental, ecological, economic and engineering systems. *Environmental Modeling and Software*, 13:105–122, 1998.

[71] P.C. Young. Data-based mechanistic modelling, generalised sensitivity and dominant mode analysis. *Computer Physics Communications*, 117:113–129, 1999.

[72] P.C. Young, P. McKenna, and J. Bruun. Identification of nonlinear stochastic systems by state dependent parameter estimation. *International Journal of Control*, 74:1837–1857, 2002.

[73] P.C. Young and M. Ratto. Statistical emulation of large linear dynamic models. *Technometrics*, pages –, 2010. tentatively accepted: available from authors.