UNIVERSITÀ DEGLI STUDI DI CATANIA
DOTTORATO DI RICERCA IN MATEMATICA APPLICATA
ALL'INGEGNERIA
XXIII CICLO

FERDINANDO CHIACCHIO

SVILUPPO DI MODELLI DINAMICI E MISURE DI IMPORTANZA PER L'ANALISI AFFIDABILISTICA DI SISTEMI COMPLESSI

TESI DI DOTTORATO

Coordinatore:
Prof. Mariano Torrisi, D.M.I.

Tutor:
Prof. Lucio Compagno, D.I.I.M.

SOMMARIO

Prefazione

Golfo del Messico, Venice (Louisiana), 20/04/2010: “Esplosione a bordo di una piattaforma petrolifera: 11 dispersi e 17 feriti”. Titola così, il 21 Aprile 2010 la prima pagina del quotidiano d’informazione il “Corriere della Sera”, per quello che sarebbe stato solo l’inizio di una delle catastrofi ambientali di matrice umana più imponenti e discusse dell’era moderna. Si tratta dell’esplosione del pozzo petrolifero e del successivo affondamento della piattaforma Deepwater Horizon a largo della costa sud-est di Venice.
(http://www.corriere.it/Primo_Piano/Esteri/2010/06/08/pop_cronistoria_marea_nera.shtml)

Di seguito le recenti vicende che hanno caratterizzato l’evoluzione della catastrofe:

- 20/04/2010, esplosione del pozzo petrolifero, localizzato a 80 km sud/est della città di Venice (Louisiana), nel golfo del Messico. Il primo bilancio parla di 126 persone coinvolte, 11 dispersi e 17 feriti di cui 4 gravi.
- 22/04/2010: affondamento della piattaforma che contiene 2,6 milioni di litri di greggio.
- 29/04/2010: dichiarato lo stato di emergenza. Vengono confermate le stime sulla perdita giornaliera che ammonta a 80.000 litri di greggio riversato in mare e l’esistenza di una terza frattura a 1500 metri di profondità che rende le operazioni di emergenza particolarmente complesse. Gli esperti iniziano a valutare le ripercussioni sull’ecosistema che si protrarrebbero per oltre 50 anni e che porteranno allo sterminio di diverse specie animali, marine e volatili.
- 30/04/2010: British Petroleum (BP), società che ha in uso la piattaforma Deepwater, dichiara di essere pronta a pagare tutte le spese legate ai danni diretti e alla bonifica delle aree coinvolte.
- 10/05/2010: BP apre un sito web a diffusione mondiale per raccogliere consigli e suggerimenti riguardanti soluzioni per chiudere le falle, bloccare il pozzo e bonificare le aree coinvolte dal disastro.
- 12/05/2010: il presidente degli USA, Barack Obama, propone una tassa di 0.1$ a carico delle compagnie petrolifere per ogni barile estratto, al fine di istituire un

fondo destinato ai programmi di ricerca per gli studi di sicurezza da perforazioni in mare.

- Dal 14 maggio cominciano le operazioni di pompaggio del greggio. Mentre BP professa un certo ottimismo sull'efficacia delle metodologie applicate, rimbalzano notizie poco incoraggianti sull'effettiva riuscita delle operazioni. Da questo momento si susseguiranno diverse missioni fallimentari; verrà anche suggerita (da uno dei quotidiani russi più venduti, il *Komsomoloskaya Pravda*) l'ipotesi dell'innesco di una carica nucleare controllata per occludere il pozzo a 1600 metri di profondità. Solo la prima settimana di giugno, l'operazione "Cut and cap" di sostituzione della valvola di sicurezza malfunzionante con un imbuto di contenimento (il Lower Marine Riser Package) viene dichiarata conclusa.
- 11/06/2010: crollano i titoli in borsa di BP.
- 15/06/2010: un fulmine colpisce uno dei battelli che partecipano alle operazioni di bonifica nel golfo del Messico.

Questi sono solo parte degli eventi seguenti l'esplosione del pozzo, documentati dai giornali.

Cosa ci insegnano questi fatti? Questi fatti insegnano che da una grave anomalia di processo (in questo caso industriale) possono avere origine tutta una serie di conseguenze a catena di incontrollabile entità. In questo caso poi si è riscontrata pure l'inadeguatezza delle misure di intervento e, soprattutto, l'incosistenza della macchina di gestione in mano ai politici, agli esperti di settore e ai grandi gruppi di potere industriale. Sono vicende come queste che obbligano tutti a delle riflessioni profonde che spaziano dalla tecnologia all'etica della stessa e che dimostrano quanto devastanti possano essere le conseguenze di valutazioni di rischio non dimensionate ai requisiti di conformità che taluni scenari di processo industriale presentano.

Nel caso della Deepwater Horizon i costi economici preventivati da BP per risarcire i danni saliranno dagli iniziali 2,5 miliardi di dollari a 20 miliardi di dollari, patteggiati con la Casa Bianca. Da un punto di vista del danno ambientale, le stime fatte non potranno mai essere definitive e sarà il tempo a mostrare le conseguenze sugli ecosistemi.

LISTA DEGLI ACRONIMI

- BDD = Binary Decision Diagram
- BE = Basic Event
- CDF = Cumulated Distribution Function
- CTMC = Continuous Time Markov Chain
- DAG = Direct Acyclic Graph
- DCS = Decision Support System
- DFT = Dynamic Fault Free
- DRBD = Dynamic Reliability Block Diagram
- DTMC = Discrete Time Markov Chain
- ExpD = Exponential Distribution of Probability
- EE = External Event
- FDEP = Functional Dependency
- FMEA = Failure Mode and Effects Analysis
- FT = Fault Tree
- FT-A = Fault Tree Analysis
- GD = Generalized Distribution of Probability
- GSMP = Generalized Semi-Markov Process
- HAZOP = Hazard and Operability Study
- MCS = Minimal Cut Sets
- MOE = Multiple Occurring Event
- MRGP = Markov Regenerative Process
- MRM = Markov Rewards Model
- (N)CTMC = (Non) Homogeneous Continuous Time Markov Chain
- PAND = Priority AND
- PE = Primary Event
- PRA = Probabilistic Risk Assessment
- RAMS = Reliability, Availability, Maintainability and Safety
- RBD = Reliability Block Diagram
- SEQ = Sequence Enforcing
- SFT = Static Fault Tree
- SPN = Stochastic Petri Net
- TE = Top Event
- UE = Undeveloped Event
- UnMOE = Non Multiple Occurring Event
- Wysiwyg = What you see is what you get

1. INTRODUZIONE

In questo capitolo vengono introdotti dei concetti propedeutici alla lettura di questo lavoro di tesi: la coppia sistema/processo, l'affidabilità, i suoi attributi e strumenti e la disciplina della Probabilistic Risk Assessment (PRA).
La conoscenza qualitativa di queste nozioni e la consapevolezza di quali siano i loro ambiti di applicazione nella vita di tutti i giorni agevola la comprensione del processo di astrazione operato dalla teoria matematica dell'affidabilità.
Per questo motivo è stato inserito anche un paragrafo dal carattere più divulgativo riguardante le origini dell'affidabilità e i campi di interesse che essa abbraccia e che l'hanno trasformata, da semplice strumento di supporto alla progettazione dei sistemi e dei processi di produzione, in una disciplina, la PRA, che detiene oramai un ruolo centrale nella pianificazione delle strategie di gestione della manutenzione, dell'operatività e della sicurezza degli impianti.

1.1 INTRODUZIONE ALLA DISCIPLINA DELL'AFFIDABILITÀ

La disciplina nota con il nome di affidabilità è stata sviluppata con lo scopo di fornire metodi per valutare se un prodotto o un servizio sarà funzionante nell'istante e per la durata in cui l'utilizzatore lo richiederà. Infatti, da un punto di vista ingegneristico e sistemistico, nulla è perfetto: nella vita reale, purtroppo, non si può aver a che fare con oggetti che rimangono privi di guasti per sempre (Modarres et al., 1999). Infine, considerazioni di ordine economico e pratico raccomandano il ragionevole impiego del concetto del design imperfetto e, di conseguenza, le problematiche che ne conseguono.
In pratica, l'affidabilità è anche una ben precisa funzione matematica; in particolare è una probabilità e non una grandezza deterministica ed è calcolabile con formule analitiche. Dato il carattere aleatorio della misura, il suo valore può essere previsto solo attraverso considerazioni di tipo probabilistico.
Nel mondo anglosassone, al fine di non confondere le due nozioni (la disciplina e la misura), l'IFIP (International Foundation for Information Processing) Working

Group WG10.4 (Dependable Computing and Fault-Tolerance) ha introdotto un nuovo termine, la *dependability*, relativo alle proprietà che descrivono la disponibilità e i fattori che la condizionano come appunto l'affidabilità (la reliability), la manutenibilità e la logistica della manutenzione (UNI, 1991). Questa differenziazione è nata nell'ambito dei sistemi ICT ed è diventata attuale in ogni settore dal momento che le moderne tecnologie dei sistemi di controllo, di supervisione ed elaborazione sono integrate nelle più varie applicazioni industriali.
La disciplina dell'affidabilità (dependability) è una teoria matematica applicata con funzioni e distribuzioni di probabilità e include gli attributi di *affidabilità*, *disponibilità*, *sicurezza* e *protezione* (WG 10.4 - Dependable Computing and Fault Tolerance, 1994).
I metodi della dependability possono essere classificati in tecniche per:

- la **previsione** (conoscere a priori quali possano essere le cause di eventi indesiderati o cosa potrebbe non funzionare);
- la **prevenzione** (conoscere ed evitare il verificarsi di tali eventi anomali o guasti);
- la **mitigazione** (attuare dove possibile misure per limitare le conseguenze di un evento indesiderato, di un mal funzionamento o di un guasto);
- la **rimozione** (predisporre in tempi rapidi gli interventi più adatti a ripristinare le condizioni di funzionamento del sistema).

Come detto in precedenza la dependability nasce nel settore ICT. Quando l'aspetto di *information* è poco significativo come nell'ambito industriale, l'albero della dependability descritto in (WG 10.4 - Dependable Computing and Fault Tolerance, 1994), è semplificabile come in Figura 1.1, senza mettere in risalto gli attributi di confidenza ed integrità, tipici dei flussi informativi.

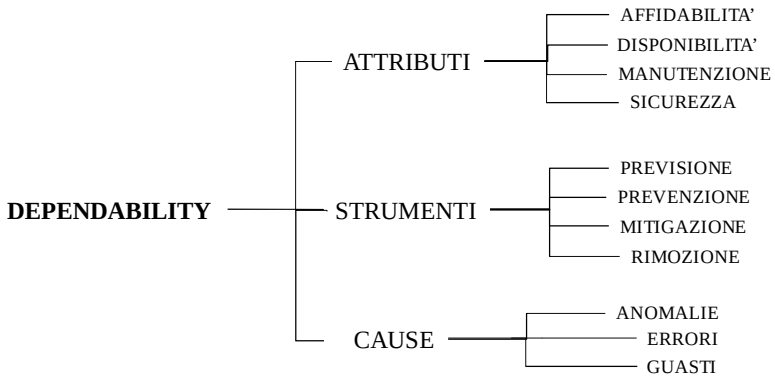


Figura 1.1: L'albero della dependability o disciplina dell'affidabilità

La valutazione degli attributi della dependability dipende dalle proprietà del sistema/processo, dalle funzioni da esso svolte e dalle condizioni operative di funzionamento (Birolini, 2003). Occorre, quindi, chiedersi:

- come deve essere adoperato il prodotto/servizio dall'utilizzatore?
- qual è la funzione che il sistema deve effettivamente svolgere?
- quali sono i valori limite delle condizioni operative ed ambientali sotto le quali il prodotto/servizio deve funzionare correttamente?
- in quale istante o intervallo di tempo il prodotto/servizio deve funzionare?

Pur non fornendo la certezza che un guasto si verifichi o meno, quindi, la teoria dell'affidabilità, applicata in modo sistematico su un sistema, fornisce risultati molto utili sui quali è possibile basare importanti decisioni sul modo in cui un impianto viene fatto funzionare, per esempio decisioni che riguardano la sicurezza.

1.2 SISTEMA/PROCESSO, COMPONENTE E MODELLO

La definizione di affidabilità è molto sensibile a ciò che viene definito sistema.
Nell'ingegneria, il termine sistema viene definito come un'entità a diversi livelli di complessità composta da persone, procedure, materiali e non, strumenti, attrezzature, hardware e software che insieme concorrono alla realizzazione di precise funzionalità (Rausand M., 2004): è un sistema ogni prodotto o servizio che viene

impiegato da un utilizzatore. Dunque, un sistema può essere sia un insieme assemblato di componenti che costituisce una parte funzionale di un'apparecchiatura, sia una sequenza di operazioni (procedura o processo) per eseguire un servizio.
Dal momento che il concetto di sistema comprende nel suo insieme sia gli aspetti fisici che di processo (Figura 1.2), dal punto di vista della dependability distinguiamo il:

- **componente:** è un oggetto (anche complesso) la cui affidabilità può essere valutata mediante i dati statistici forniti dai produttori;
- **sistema**: è un apparato fisico complesso e ben strutturato, composto da componenti;
- **processo:** corrisponde alle funzionalità svolte dal sistema.

Per esempio, per una raffineria di petrolio possiamo dire che gli impianti corrispondono al sistema mentre le funzioni implementate, per esempio le procedure chimico-fisiche di raffinazione del greggio, sono i processi ad essi associati.
Dal momento che la qualità finale dei risultati esposti da un sistema dipendono anche da quelli del processo che li realizza, è chiaro che le due entità, sistema e processo, debbano essere considerate alla stessa stregua.

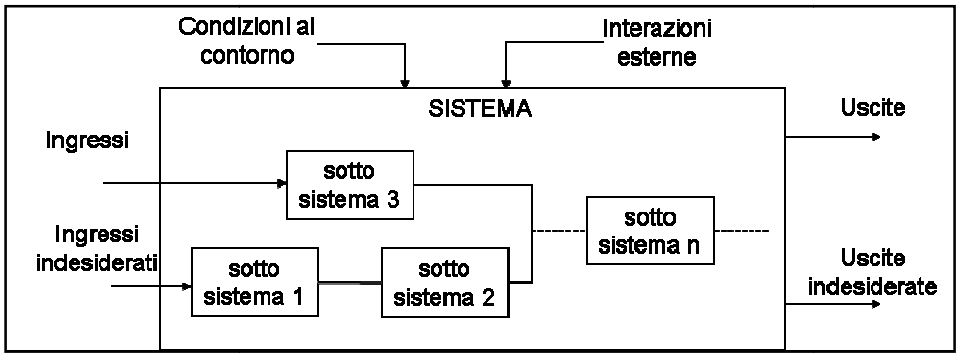


Figura 1.2: rappresentazione schematica di un sistema generico

Lo studio di un sistema/processo è un'attività importante perché mediante esso possono venir fatte delle previsioni sulle performance offerte e possono essere proposte soluzioni migliorative per favorire il raggiungimento degli obiettivi preposti.

Il metodo più semplice per valutare la qualità del risultato di un sistema/processo è dato dalla misura degli output del sistema e dal loro confronto con i risultati attesi. Questo tipo di approccio non è scorretto ma, indubbiamente, porta con sé dei limiti inaccettabili di sicurezza, di fattibilità ed economici. Infatti, se supponessimo di poter valutare dei miglioramenti solo grazie a misure dirette dei risultati di sistema/processo, per ogni tentativo o per ogni miglioria prevista si dovrebbe attendere l'esito della nuova prova su cui mettere a punto gli ulteriori raffinamenti.
I **modelli** servono per ovviare a questo problema. Infatti, un modello non è altro che un'astrazione del sistema/processo o di alcune sue parti o funzioni su cui possono essere condotte misure e prove ripetibili.
In generale, esistono diversi tipi di modelli la cui natura dipende dal sistema/processo che descrivono e dal tipo di elaborazioni a cui vanno sottoposti.
Per le applicazioni di sicurezza, di affidabilità e di manutenzione si fa uso dei modelli stocastici.

1.3 CLASSIFICAZIONE DEI GUASTI

Il termine **guasto** indica la "cessazione dell'attitudine di un'entità a eseguire la funzione richiesta" ovvero una variazione delle prestazioni del dispositivo che lo renda inservibile per l'uso al quale esso era destinato (UNI, 1991):

1. a seguito del guasto di un'entità, questa entità è in avaria;
2. il guasto è un evento, passaggio da uno stato ad un altro, mentre l'avaria è uno stato;
3. la definizione di guasto non si applica a un'entità che consiste unicamente di una struttura programmata (software).

Una definizione alternativa che mette in luce il legame con il concetto di rischio è quella fornita in (Modarres et al., 1999) in cui, con un'accezione estremamente generale, si giustifica l'esistenza del guasto mediante il concetto della *sorgente di sfida*. Una sorgente di sfida è l'insieme delle prove e dei rischi a cui il sistema si sottopone per ottenere dei risultati. Durante la sfida, condizioni esterne o interne al

sistema possono determinare il fallimento della prova cioè il guasto del sistema o di una sua parte.

Per quanto intuitiva sia la definizione, i guasti possono essere classificati in funzione degli effetti che hanno su un sistema e sulle modalità con cui agiscono. In questi termini, risulta guasto anche un dispositivo che non esegue correttamente la funzione per la quale è stato progettato. Per esempio una stampante che non stampa è certamente guasta, ma si può ritenere guasta anche una stampante che stampa i caratteri deformandoli o sporcando i fogli (qualità).

Dal punto di vista dell'operatività si possono distinguere:

- **guasti parziali**: determinano una variazione delle prestazioni del dispositivo tale da non compromettere del tutto il funzionamento (degrado delle prestazioni o perdita di qualità del prodotto);
- **guasti totali**: causano una variazione delle prestazioni del dispositivo tale da impedirne del tutto il funzionamento.

L'ipotesi generale è che al tempo t=0 un dispositivo sia completamente funzionante e privo di difetti. Dal punto di vista temporale si possono distinguere:

- **guasti intermittenti**: dovuti ad una successione casuale di periodi di guasto e di periodi di funzionamento, senza che ci sia alcun intervento di manutenzione (esempio tipico il blocco di funzionamento di un computer che riprende a funzionare dopo che viene spento e riacceso);
- **guasti progressivi:** dovuti ad un graduale cambiamento nel tempo di determinate caratteristiche dell'entità.
- **guasti continui:** per essere rimosso occorre un intervento manutentivo di riparazione/sostituzione.

Occorre precisare che la condizione di guasto si riferisce in generale al solo dispositivo preso in esame: se tale dispositivo è inserito in un sistema più complesso, il suo guasto può anche non causare il guasto dell'intero sistema, pur avendo effetti negativi sulla sua affidabilità. Ad esempio, un guasto meccanico al motore rende inservibile un'automobile mentre se si guasta il tachimetro l'automobile continua a funzionare, anche se non riusciamo a sapere a che velocità stiamo procedendo.

Anche in questo caso possiamo allora distinguere:

- **guasti di primaria importanza**: quelli che riducono la funzionalità dell'intero sistema del quale fanno parte;
- **guasti di secondaria importanza**: quelli che non riducono la funzionalità dell'intero sistema del quale fanno parte;
- **guasti critici:** ancora più gravi dei guasti di primaria importanza, sono quei guasti che rappresentano un rischio per l'incolumità delle persone. Ad esempio, il guasto dell'impianto frenante di un autoveicolo può mettere in serio pericolo la vita dei passeggeri e può, quindi, considerarsi critico.

Un'altra classificazione dei guasti può essere fatta in funzione della loro distribuzione nella vita dei componenti che colpiscono. Si hanno:

- **guasti infantili**: avvengono nel primo periodo di vita dei componenti (periodo di rodaggio) e la loro frequenza di accadimento decresce gradualmente nel tempo. La natura di questi guasti è legata a difetti intrinseci dei componenti che non sono emersi durante i collaudi; in presenza di una buona progettazione sono dovuti essenzialmente ad errori di costruzione e, principalmente, di montaggio; il periodo di questo tipo di guasti può variare da poche decine ad alcune centinaia di ore di funzionamento;
- **guasti casuali**: sono quelli che si verificano durante l'intera vita utile dei componenti ed hanno una frequenza di accadimento indipendente dal tempo; sono dovuti a fattori che neanche un buon progetto ed una buona esecuzione possono eliminare;
- **guasti per usura**: sono quelli che si verificano nell'ultimo periodo di vita di componenti e sono dovuti a fenomeni di invecchiamento e deterioramento dei componenti; per ciò la loro frequenza di accadimento cresce al passare del tempo.

Se consideriamo una popolazione di componenti nuovi, tutti uguali, non riparabili e li facciamo funzionare nelle medesime condizioni operative ed ambientali a partire dallo stesso istante t=0 è possibile tracciare il diagramma mostrato in Figura 1.3, che

descrive l'andamento del tasso di guasto istantaneo[1] (ossia una misura della frequenza di accadimento) in funzione dell'età dei componenti.

Tale funzione rappresenta la frequenza con la quale si guastano i componenti e si misura in numero di guasti (rapportato al numero di componenti ancora in vita) per ora di funzionamento. Il diagramma di Figura 1.3 assume una caratteristica forma a "vasca da bagno" che consente di visualizzare in modo chiaro la precedente classificazione in guasti infantili, casuali e per usura.

Il periodo dei guasti infantili corrisponde al tratto iniziale della curva (periodo di rodaggio) al quale corrisponde un tasso di guasto decrescente: la frequenza dei guasti, che è inizialmente elevata perché si guastano tutti quei componenti che risultano "deboli" (errori di costruzione o di montaggio), tende a decrescere rapidamente e si stabilizza su un valore minimo. Questo valore minimo del tasso di guasto si mantiene pressoché costante per un intervallo di tempo al quale si dà il nome di "vita utile", caratterizzato solo da guasti di tipo casuale.

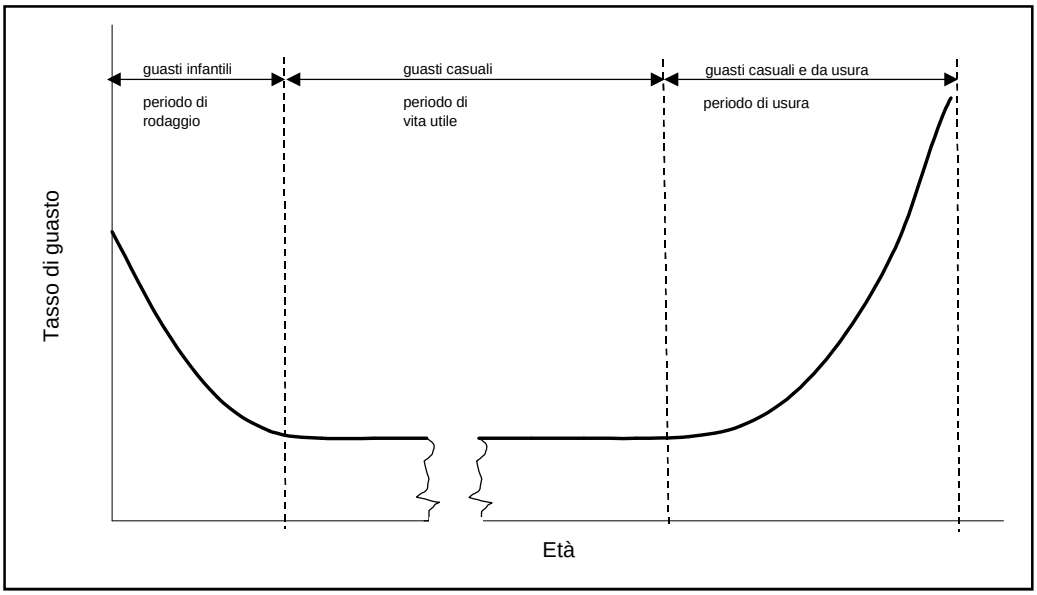


Figura 1.3: tasso di guasto dei componenti in funzione dell'età

Il periodo di vita utile dei componenti si può considerare concluso quando cominciano ad intervenire fenomeni di usura, a causa dei quali la frequenza dei

[1] la definizione del tasso di guasto sarà data nel seguito

guasti tenderà ad aumentare mettendo rapidamente fuori uso tutti i componenti sopravvissuti ai precedenti periodi di esercizio.

1.4 OBIETTIVI DELLA DISCIPLINA DELL'AFFIDABILITÀ

In ambito industriale occorre garantire la continuità di funzionamento degli impianti di produzione, la qualità dei prodotti ed il funzionamento in sicurezza sia degli impianti sia dei prodotti.
Per chiarire meglio come l'affidabilità possa aiutarci nel conseguire tali scopi possiamo analizzare la storia della nascita e della evoluzione di questa disciplina.
Le prime tracce di studi di affidabilità si ebbero tra le due guerre mondiali in campo aeronautico: si doveva decidere quale fosse la migliore configurazione per il sistema di propulsione degli aerei a più motori. Questi studi però inizialmente ebbero carattere prettamente sperimentale così come sperimentali erano anche i dati sulla frequenza di guasto di apparecchiature che si trovavano a bordo degli aerei, espressa in termini di numero medio di sostituzioni della stessa apparecchiatura. Intorno al 1930 questi dati cominciarono ad essere elaborati statisticamente, fornendo utili indicazioni sui miglioramenti da apportare ai progetti.
Tra il 1943 ed il 1950 sia i tedeschi (Von Braun) sia gli americani che operavano in ambito militare, avendo constatato che i malfunzionamenti avevano effetti negativi di notevole entità sia sull'operatività sia sui costi di mantenimento dell'apparato bellico, cercarono di dare una soluzione ingegneristica ai problemi affidabilistici. I missili tedeschi V1 e V2 furono i primi sistemi sui quali venne applicato con successo il concetto di affidabilità di sistema, partendo dall'affidabilità dei singoli componenti.
Tutti questi studi sfociarono nel 1952 nella definizione di affidabilità come "la probabilità che un oggetto adempia alla sua specifica funzione per un tempo determinato e sotto determinate condizioni".
La diffusione della disciplina dall'ambito militare a quello civile si ebbe intorno agli anni '60 a mano a mano che in tutti i settori i sistemi divenivano sempre più complessi ed automatizzati.

Alla fine degli anni ’80 gli studi affidabilistici entrarono a far parte del TQM (Total Quality Management) ed alcuni metodi di valutazione dell’affidabilità dei sistemi cominciarono ad essere richiesti per ottenere la certificazione di qualità ISO-9000.
Dalle brevi note storiche sull’origine dell’affidabilità si può intuire come il campo di interesse di tale materia si sia via via ampliato, trasformandola da semplice strumento di supporto alla progettazione ed alla produzione dei sistemi/processi in una disciplina che ha ormai assunto un ruolo centrale nella visione più moderna della progettazione all’interno della quale vengono considerati prioritari ed integrati gli aspetti legati alla sicurezza, alla qualità ed ai costi.

1.4.1 SICUREZZA

L’analisi di affidabilità risulta, come è ovvio, particolarmente utile in quelle tipologie impiantistiche che utilizzano sostanze pericolose (impianti soggetti a rischi di incidenti rilevanti, che possono coinvolgere anche aree adiacenti agli stabilimenti produttivi), per valutare la probabilità che il guasto di un componente o di un sistema di sicurezza possa determinare una sequenza incidentale con gravi conseguenze sulla incolumità delle persone.
Anche in impianti che non sono soggetti a rischi di incidente rilevante un’analisi di affidabilità può avere benefici effetti sulla sicurezza, per esempio per garantire l’incolumità del personale addetto ad operazioni critiche (sostanze pericolose o macchine particolari) o per valutare l’affidabilità delle procedure operative normali e di quelle di emergenza.

1.4.2 QUALITÀ

La scelta di un bene o servizio tra diverse soluzioni è dettata in generale dalla valutazione del rapporto tra la sua qualità ed il suo costo.
In effetti, se si cerca di definire un prodotto “di qualità” è spontaneo considerare, tra le caratteristiche che il prodotto deve possedere, anche la durata (per quanto tempo si può utilizzare effettivamente il componente), l’affidabilità (con quale frequenza si

guasta il prodotto), la manutenibilità (quanto facilmente il prodotto può essere riparato).

Se la qualità viene, quindi, intesa in termini di adeguatezza del bene allo scopo al quale è destinato, alla sua determinazione contribuiscono principalmente due fattori:

- **conformità**, che tiene conto della aderenza delle prestazioni alle specifiche progettuali e/o commerciali;
- **affidabilità**, che tiene conto della capacità del prodotto/servizio di mantenere le sue caratteristiche di funzionamento nel tempo.

1.4.3 COSTI

Il costo annuo totale delle misure di riduzione del rischio comprende:

- **costi di investimento** (per esempio, acquisto nuove apparecchiature di sicurezza), da ammortare in un certo periodo di tempo, con un certo tasso;
- **costi di manutenzione** delle apparecchiature stesse;
- **costi operativi** (per esempio, per l'aggiunta di personale o per l'addestramento dello stesso).

Normalmente non vengono inclusi ulteriori costi operativi per la realizzazione di procedure operative più sicure, in quanto si assume che questi vengano già considerati tra i costi di realizzazione dell'intervento.

Questi costi vengono, in genere, valutati in funzione dell'affidabilità richiesta al sistema in esame, in quanto questa può essere ottenuta con due diverse strategie:

1. richiedendo al fornitore un prodotto con affidabilità molto elevata; questo comporta costi di progettazione e di produzione elevati e, quindi, un costo d'acquisto piuttosto elevato ma minori costi di manutenzione (parti di ricambio e manodopera);
2. richiedendo al fornitore un prodotto di affidabilità inferiore e, quindi, di costo inferiore ma prevedendo un adeguato programma di manutenzione con un aumento dei costi di manutenzione.

Secondo una visione tradizionale il costo totale minimo si ottiene quando i costi di fornitura e quelli di manutenzione si bilanciano, ovvero quando c'è equilibrio tra il

costo dei programmi di sicurezza/affidabilità ed i costi di mancata produzione da sostenere in caso di guasto/incidente, come mostra la Figura 1.4.

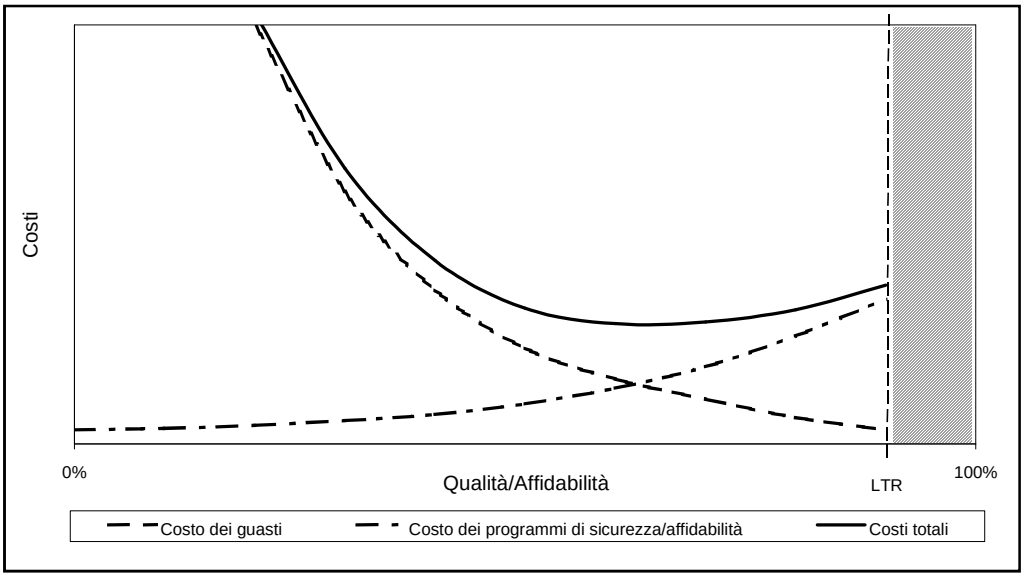


Figura 1.4: Costi della sicurezza/affidabilità (visione tradizionale)

In Figura 1.4, si è tracciata una retta verticale che rappresenta il Limite Tecnicamente Raggiungibile (LTR) ovvero quel livello di qualità/sicurezza oltre il quale non è opportuno spingersi per ragioni tecniche: si vuole infatti sottolineare che la sicurezza totale è impossibile da raggiungere e che l'attuale livello tecnologico, già piuttosto avanzato, consente normalmente solo piccoli miglioramenti sui progetti, mentre non ha senso spingersi troppo oltre nell'adottare sistemi di sicurezza che potrebbero rivelarsi controproducenti, in virtù della complessità progettuale e funzionale alla quale conducono.

Una visione più moderna del problema suggerisce, invece, che i costi legati ai guasti divengono molto più elevati se nascono questioni di sicurezza e se si considerano in essi anche fattori difficilmente quantificabili come la vita umana, i costi di inquinamento dell'ambiente e la perdita d'immagine dell'azienda. In quest'ottica i costi dei programmi di sicurezza divengono dei benefici figurativi in quanto determinano dei "mancati costi", cioè fanno sì che in caso di incidente non si debbano sostenere costi ben maggiori. L'andamento tipico delle curve costo/sicurezza, secondo l'ottica moderna è mostrato in Figura 1.5.

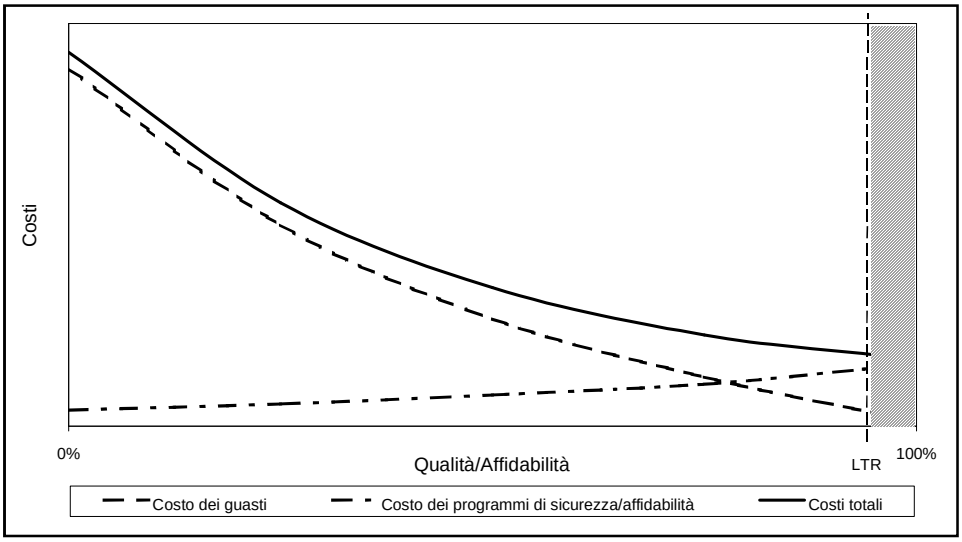


Figura1.5: Costo della sicurezza/affidabilità (visione moderna)

L'analisi affidabilistica fornisce risultati utili in qualunque momento essa venga eseguita, anche se uno studio effettuato sin dalle fasi progettuali consente ovviamente di realizzare interventi molto più efficaci (rapporto costi/benefici più basso).

In fase di progettazione si possono, infatti, individuare i punti deboli del progetto ed i componenti critici del sistema, cioè quelli che influenzano maggiormente l'affidabilità del complesso, e ciò ci consente di scegliere componenti più affidabili o meglio ancora di configurare il sistema in modo da rendere l'affidabilità del complesso meno dipendente dall'affidabilità del componente critico (ridondanza).

In fase di esercizio, invece, si può stabilire una strategia di manutenzione che riduca al minimo i tempi di fuori servizio del sistema (costi di mancata produzione) ed i costi di manutenzione.

Possiamo quindi concludere che le analisi di affidabilità rappresentano gli studi quantitativi, sia pure in termini probabilistici, da eseguire non solo per realizzare corrette analisi del rischio dei sistemi e soddisfare quindi eventuali adempimenti richiesti dalle normative vigenti, ma anche per ottenere prodotti di qualità che risultino competitivi in mercati che divengono sempre più esigenti.

1.5 PROBABILISTIC RISK ASSESSMENT

L'applicazione sistematica e formalizzata delle tecniche dell'affidabilità hanno dato vita alla cosiddetta “valutazione probabilistica di rischio” (PRA) che si è andata affermando inizialmente in seno alle attività di analisi di rischio e di manutenzione tipiche degli impianti di produzione di energia nucleare e, successivamente, per gli impianti/sistemi complessi con implicazioni di sicurezza.

Lo scopo della PRA (Office of Safety and Mission Assurance, 2002) è di identificare e valutare quali siano i possibili rischi associati a sistemi tecnologicamente complessi in modo da proporre le migliori soluzioni costi-benefici tali da migliorare la sicurezza e, più in generale, le performance del sistema. L’identificazione logica degli eventi indesiderati viene effettuata secondo due modalità di indagine:

- quella induttiva che dalle cause iniziali risale all'evento finale, delineando gli scenari evolutivi mediante la verifica di condizioni di successo o fallimento degli eventi individuati;
- quella top-down (tipica dei fault tree) che, partendo da un evento finale (il Top Event), muovendo indietro nel tempo, ne scova le cause (Pham, 2003), (Xing & Amari, 2008).

Dunque, uno dei risultati essenziali della PRA è fornire delle informazioni su come le parti di un sistema/processo operano e sulle interazioni che li caratterizzano.

La valutazione qualitativa può essere condotta partendo dalle tre informazioni fondamentali che riguardano le anomalie del sistema/processo (Figura 1.6):

a. quali sono le anomalie che possono caratterizzare un sistema;
b. quali sono le combinazioni di tali anomalie che si traducono in eventi iniziatori;
c. quali sono le conseguenze a cui si va incontro.

Come si può intuire dallo schema di Figura 1.6, gli esperti devono identificare quali possano essere le anomalie associate al sistema/processo e le loro mutue interrelazioni. Alle anomalie individuate bisogna associare delle cause, gli eventi iniziatori che possono essere legati a guasti dei componenti, a errori umani, a condizioni operative non previste e gestite dal sistema/processo.

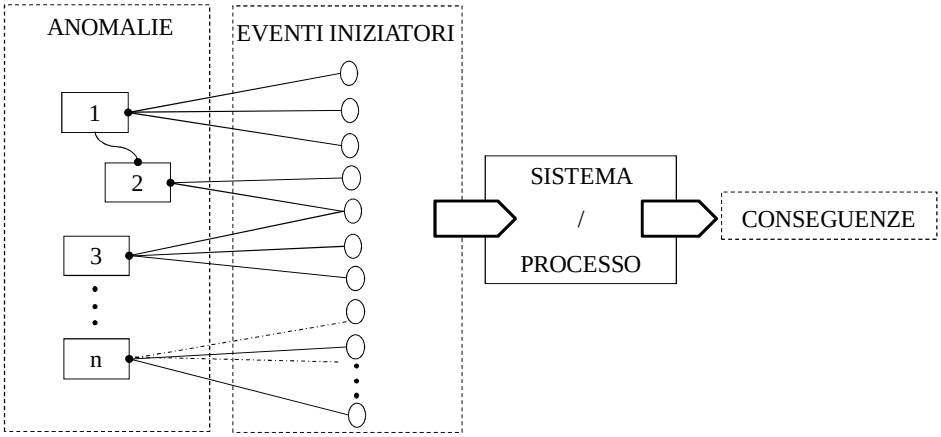


Figura 1.6: Modello semplificato di diagnostica qualitativa del PRA

Queste cause possono avvenire con un ordine temporale diverso che, a sua volta, può essere determinante nella reale concretizzazione dell'anomalia. A seconda della metodologia utilizzata, anche la gestione della catena temporale può essere o meno modellata e, quindi, valutata. I sistemi che si prestano a questo tipo di modellazione sono oggi definiti modelli dinamici. La valutazione delle conseguenze conclude la prima fase dello studio della PRA. Generalmente le conseguenze individuate possono essere la descrizione di altri scenari di rischio interni o esterni al sistema/processo, la stima dei costi e dei tempi di ripristino delle condizioni normali, la valutazione dei danni fisici a macchine o persone coinvolte, ecc.

Quando la descrizione qualitativa del rischio è completa, la successiva fase della PRA riguarda la quantificazione associata alle anomalie individuate, la cosiddetta *risk integration* (integrazioni quantitativa di rischio, Figura 1.7).

Le principali valutazioni che emergono dall'uso combinato delle metodologie della PRA riguardano la:

- determinazione degli elementi che contribuiscono maggiormente ai rischi del sistema/processo;
- le incertezze associate a tali stime;
- l'efficacia delle varie strategie di mitigazione/ottimizzazione disponibili.

Da un punto di vista operativo, il vantaggio offerto dalla PRA è quello di fornire un metodo sistematico a disposizione dell'esperto del rischio per la modellazione del sistema/processo. Con l'ausilio delle tecniche della PRA, le ambiguità del processo

di analisi del rischio, delle fasi che lo costituiscono e delle possibili metodologie di modellazione in forza agli esperti del rischio vengono ridotte.

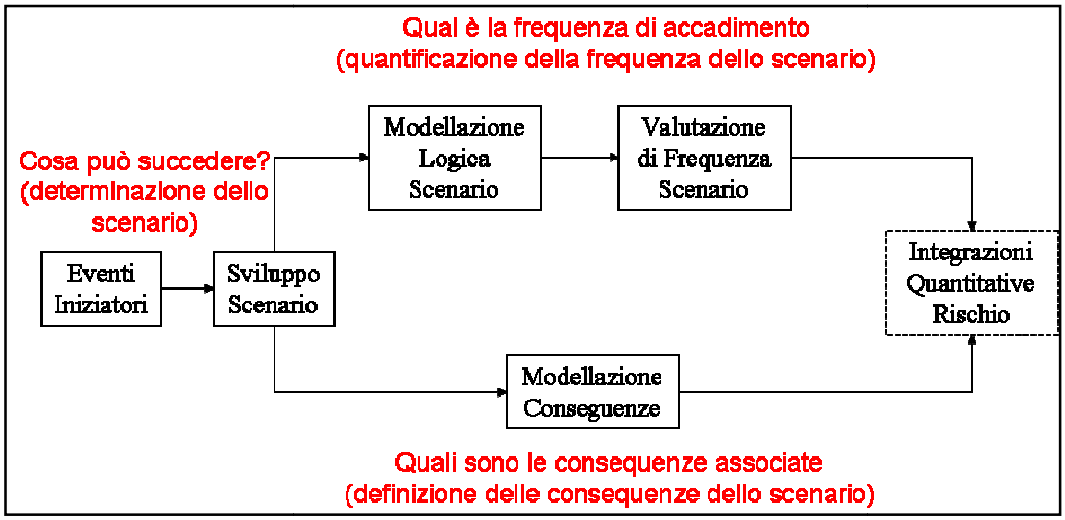


Figura 1.7: costruzione del rischio associato ad uno scenario

Le fasi che compongono la PRA (Figura 1.8) sono varie e possono essere implementate in modo diverso a secondo del sistema/processo sotto analisi. Alcune fasi (in bianco) divengono auspicabili qualora il sistema/processo in esame fosse caratterizzato da criticità che rendono necessarie delle fasi di verifica preliminare (alcuni aspetti della familiarizzazione e raccolta informazione e la validazione degli eventi iniziatori). L'attività di valutazione delle performance, di classificazione e sensitività dei alcuni indicatori è supportata solo nei casi in cui il sistema è dotato delle tecnologie che consentono la memorizzazione dei dati di processo (come per esempio accade nei sistemi informatici in rete, per i quali è possibile acquisire uno storico di informazioni sui dati di traffico, sui tempi di servizio, ecc.).

Come mostra la Figura 1.8, la PRA presenta un flusso logico ben preciso che favorisce l'approccio sistematico all'attività di valutazione del rischio.

Alla base della PRA vi è la produzione di una documentazione specifica per ogni fase. In questo modo viene garantito l'assolvimento dei vari punti di ogni fase, il risultato delle singole attività e l'individuazione degli attori responsabili di queste analisi. Le prime fasi della PRA sono quelle di carattere investigativo e qualitativo, in cui vengono definiti i contorni entro i quali sviluppare le successive valutazioni di modellazione e quantificazione (Modarres, 2008) .

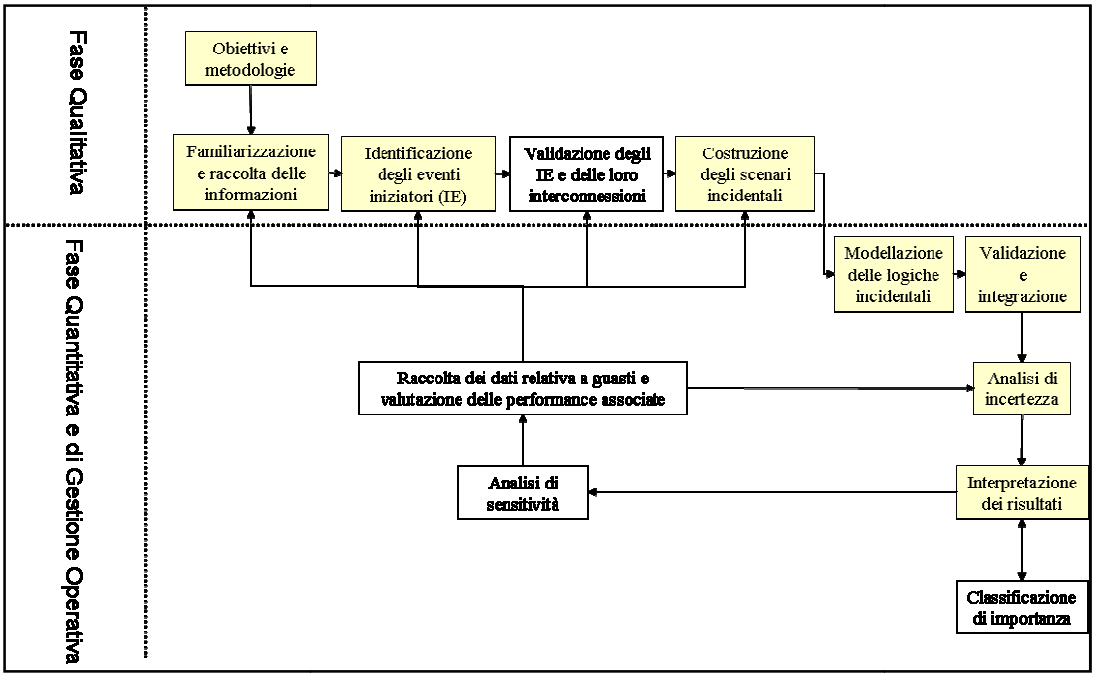


Figura 1.8: attività che compongono la PRA. In giallo le attività standard tipiche di tutte le analisi di rischio

Nella fase denominata “obiettivi e metodologie”, gli attori dell’analisi devono definire quali sono gli obiettivi della PRA. Questi possono comprendere obiettivi quali la sicurezza, la qualità, la continuità di esercizio, il controllo dei costi ecc. Per raggiungere tali obiettivi possono essere pianificate molteplici attività che comprendono la revisione dei cicli manutentivi, dei turni lavorativi, del supporto alle decisioni e delle responsabilità. Noti gli obiettivi diventa possibile fare un inventario delle metodologie per ogni attività. Queste possono comprendere le tecniche di analisi, l'individuazioni di indicatori (affidabilità, disponibilità), la classificazione degli obiettivi più critici, ecc.

Quando si ha esperienza del processo/sistema sotto studio è possibile avviare la fase di “familiarizzazione e raccolta delle informazioni”. Quest’attività può essere gestita mettendo insieme il know-how tecnologico e l’esperienza degli analisti, insieme con le analisi storiche riguardanti sistemi/processi simili a quello sotto studio. Generalmente il know-how tecnologico viene messo a disposizione dai fornitori dei componenti o dai loro utilizzatori. Lo studio dei layout fisici (P&IDs, Process and Instrumentation Drawings) è necessario per l’attività di modellazione che verte sulla conoscenza delle interconnessioni fisiche fra le parti del sistema, generalmente

riassunte in una matrice delle dipendenze. Attraverso i layout fisici deve anche essere possibile identificare i sistemi di sicurezza, di controllo e di mitigazione che comprendono le parti automatiche e quelle gestite da personale qualificato.

La fase di "identificazione degli eventi iniziatori" è caratterizzata dalla classificazione dei possibili eventi esterni o interni al sistema/processo, da cui possono avere origine episodi anomali o indesiderati. Essa viene implementata mediante l'uso di metodologie qualitative quali l'analisi HAZOP (Hazard & Operability), la decomposizione funzionale o FMEA (Failure Mode and Effect Analysis), consolidate per l'analisi di rischio dei processi industriali.

Il sistema/processo viene analizzato nelle diverse modalità operative che lo caratterizzano. Tali modalità vengono realizzate da diverse parti del sistema che assolvono a funzioni specifiche. Fintanto che i parametri del sistema rimangono all'interno dei range di tolleranza (in quella che viene chiamata *modalità operativa normale*) non si prevedono situazioni anomale. Tuttavia, durante questa modalità possono sopraggiungere condizioni di funzionamento anormali che, in un primo momento, vengono considerate di transizione. Da questo momento in poi sono possibili due evoluzioni: nella prima il sistema è capace di tornare nella modalità operativa normale per effetto dei sistemi di mitigazione e sicurezza tipici del processo. Nella seconda, invece, lo stato anomalo si protrae e possibilmente diffonde il proprio effetto su altre parti del sistema/processo; per effetto di tale anomalia, l'evento cha ha causato l'iniziale disallineamento dalla modalità operativa normale può essere definito come un evento iniziatore (IE).

Una volte noti gli eventi iniziatori e le loro relazioni si procede con la costruzione degli scenari incidentali. L'obiettivo di questa fase è di descrivere con precisione l'evoluzione degli scenari e le potenziali problematiche associate. Uno degli strumenti più noti per questa attività è la rappresentazione mediante gli Event Tree. Gli Event Tree possono descrivere in maniera cronologica, mediante una formalizzazione estremamente chiara e concisa il successo o il fallimento delle manovre di mitigazione richieste in risposta a dei determinati eventi iniziatori.

In generale, l'ordine di procedure che riguarda questa fase della PRA vede:

- l'identificazione delle misure di mitigazione proprie di ogni evento iniziatore;

- l’identificazione delle procedure manuali ed automatizzate (mediante sistemi software ed hardware) associate alle misure di mitigazione e le ipotesi di attuazione;
- lo sviluppo dei diagrammi di event-tree tali da rendere chiare le possibili evoluzioni legate agli eventi iniziatori.

Il quadro ipotizzato dall’evoluzione degli scenari mediante event tree può essere approfondito nella fase di modellazione logica. L’uso di modelli stocastici è la soluzione più adottata in questo ambito della PRA perché in tal modo è possibile calcolare la probabilità di occorrenza di un evento. Il campo della modellazione è molto impervio perché la complessità che un modello può raggiungere può rendere la sintesi di un risultato impraticabile (Epstein & Rauzy, 2004). Da un lato, dunque, il trade-off tra la precisione di un modello e la sua risoluzione obbliga gli analisti del rischio a prendere in considerazione solo modelli risolvibili. Questa proprietà viene spesso verificata cercando di concentrare lo studio di un sistema su parti indipendenti o fissando un limite al contorno per le possibili dipendenze che un evento presenta. In questo modo l’esito della sintesi offre dei risultati valutabili, su cui poter basare ulteriori indagini. Dall’altro lato, tuttavia, un modello semplice da risolvere può risultare poco realistico o limitato. Per questo motivo, le diverse tecniche di modellazione in forza alla PRA possono essere usate sinergicamente per due motivi principali: il primo è quello di analizzare uno scenario e mettere a confronto i risultati di diverse metodologie; il secondo è quello di realizzare un modello ibrido (descritto da tecniche diverse) che, a seconda della parte di sistema, utilizza la tecnica di modellazione più adatta.

Affinché i modelli stocastici siano credibili, diventa necessario alimentare i loro ingressi con dati aventi una forte fondazione statistica. Per questo il miglior modo per effettuare valutazioni di rischio per un sistema/processo è quello di utilizzare informazioni collezionate nella storia di sistemi e processi affini. Esistono database di dati per ogni tipo di componente impiantistico e anzi, in conformità alle norme di qualità, i fornitori hanno l’obbligo di fornire, all’interno dei datasheet, le misure di affidabilità medie per i componenti che producono e commercializzano. Tipicamente i dati necessari includono i tempi di guasto, di riparazione, di test e di down-time.

Noti gli ingressi del modello stocastico, la risoluzione fornirà i valori di probabilità di accadimento dell'evento modellato. Chiaramente, questo processo varia a seconda del modello e della sua complessità.

La fase di analisi delle incertezze è una parte significativa della PRA. Qualsiasi calcolo è affetto da incertezze che, sebbene di modesta entità, propagandosi internamente al modello, possono degenerare in significative perdite di consistenza per i risultati. Nella PRA le incertezze sono codificate sotto forma di distribuzioni di probabilità. Per esempio, la probabilità di guasto di un sistema dovrebbe essere rappresentata da una distribuzione di probabilità dei tempi di guasto che mostra il range entro il quale il guasto è più probabile. Dunque, quest'attività della PRA deve caratterizzare più eventi iniziatori possibile. Altre cause di incertezze provengono dalla modellazione mediante i modelli stocastici. Questo è ciò che accade, per esempio, nell'adozione di modelli ibridi gestiti mediante la tecnica della gerarchizzazione; la tecnica, come vedremo, può introdurre significative approssimazioni (Modarres, 2008). Dunque, sebbene l'obiettivo della PRA è di fornire delle informazioni utili per gestire al meglio le situazioni di rischio e offrire delle soluzioni per il miglioramento dei sistemi, questa fase della PRA deve servire per mettere nero su bianco i limiti delle soluzioni modellistiche adottate. L'adozione della pratica conservativa – che si fonda su valutazioni pessimistiche e che è spesso usata nell'attività di quantificazione dei rischi – senza un'adeguata documentazione di incertezza non si incastra adeguatamente con le successive attività di ottimizzazione previste dalla PRA. I risultati delle valutazioni di incertezza possono essere ottenuti mediante delle simulazioni Montecarlo che, mediante grafici di distribuzioni probabilistiche, mostrano la dispersione dei valori di rischio dovuta alle caratteristiche stocastiche delle grandezze introdotte.

Le successive fasi della PRA riguardano l'analisi di sensitività e la classificazione, in ordine di importanza, della componentistica del sistema e delle funzioni che concorrono alla realizzazione del processo. Queste valutazioni vengono effettuate mediante gli strumenti matematici della sensitività e delle misure di importanza. La sensitività è una misura differenziale che esprime la variazione di una grandezza (per esempio l'affidabilità di un sistema) rispetto a piccole oscillazioni di un parametro, attorno al valore nominale di riferimento. Le misure di importanza, invece, sono per

lo più definite per gli elementi finiti del sistema, i componenti che lo costituiscono; questi elementi devono essere localizzabili nella struttura fisica del modello.

Per quanto riguarda le procedure di analisi della sensitività, esse possono essere fatte in modo da non considerare gli effetti di variazioni combinate di più parametri (in quanto la renderebbero estremamente più complicata). Isolando i parametri uno per volta si cerca di ottenere un quadro di massima sull'influenza che essi hanno sul valore calcolato.

Per quanto riguarda il calcolo delle misure di importanza, data la complessità strutturale dei sistemi in esame, una pratica molto utilizzata è quella di studiare il sistema/processo in modo da scomporre quest'ultimo in sottosistemi. La scomposizione in sottosistemi può esser fatta per funzionalità o individuando cluster di componenti altamente interconnessi. In questo modo, il calcolo delle misure di importanza viene incentrato sui sottosistemi piuttosto che sui singoli elementi del sistema. Il vantaggio di questa tecnica è di individuare i sottosistemi più delicati che, in un momento successivo, possono essere studiati in maniera più approfondita.

Le attività della PRA concorrono, dunque, alla definizione di un quadro di risk assessment che, in ultima analisi, ha la necessità di essere compreso e validato. L'interpretazione dei risultati è fondamentale per determinare se l'attuale piano di intervento, i dispositivi di sicurezza e la gestione degli orizzonti manutentivi sono dimensionati adeguatamente per far fronte alle anomalie del sistema/processo. In questo modo possono essere pianificate strategie di diverso ordine temporale per la definizione degli interventi atti a migliorare la qualità implementata.

Come già detto, l'implementazione della PRA all'interno di una realtà sistema/processo può avere diversi gradi di complessità. Il denominatore comune deve rimanere la standardizzazione prevista dalle logiche con cui le fasi della PRA si susseguono, permettendo agli esperti del rischio di scegliere come meglio approfondire le analisi che costituiscono le varie fasi. Questo ultimo concetto è sintetizzato nel diagramma di flusso circolare del CRM (Continuous Risk Management Process, Figura 1.9) (Stametalos M. (OSMA), 2002), adottato alla NASA. Questa pratica fornisce un approccio disciplinato e documentato alla gestione dei rischi mediante un ciclo di progettazione proattivo che riguarda le attività di decision making relative all'azienda:

1. valutazione in essere dei rischi che possono verificarsi;
2. determinazione delle priorità dei rischi;
3. implementazione delle strategie di mitigazione;
4. valutazione delle strategie implementate;
5. controllo e miglioramento delle strategie.

Figura 1.9: il processo ciclico di gestione del rischio (Stametalos M. (OSMA), 2002)

Il CRM per sua natura, impone la cooperazione di personale con diverso background ai vari livelli dell'organigramma (tecnici, esperti, ricercatori, manager, risorse esterne) promuovendo anche un uso efficiente delle risorse umane coinvolte e localizzando le responsabilità ai vari livelli. La Figura 1.10 schematizza queste dipendenze e semplifica la rappresentazione ad albero (Modarres et al., 1999).

Grazie all'approccio ciclico, nella fase di identificazione e analisi vengono ridisegnati i possibili miglioramenti sinergicamente alle funzioni di ricerca e sviluppo mediante l'applicazione di nuove tecnologie. In questo modo viene permesso il superamento delle logiche e dei vincoli incontrati nelle precedenti analisi e l'implementazione di soluzioni più efficaci e/o più efficienti.

Anche nel campo industriale, l'ingegnerizzazione dei processi e l'esperienza collezionata nella messa in opera dei sistemi industriali ha permesso il miglioramento delle attività della PRA.

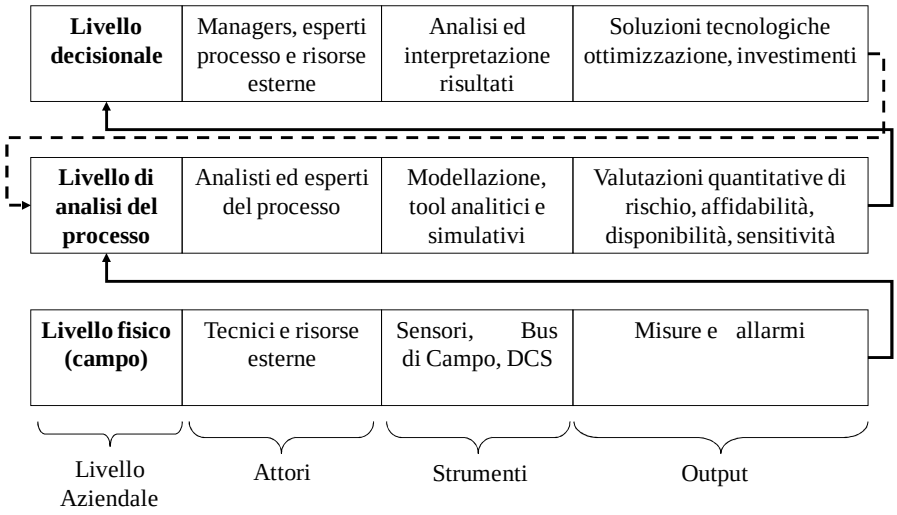


Figura 1.10: evoluzione della PRA con riferimento ai livelli aziendali, agli attori e agli strumenti associati

Negli ultimi anni, la tecnologia basata sulla misurazione e la condivisione in tempo reale dei dati provenienti dal campo ha chiuso il loop fra gli elementi che costituiscono la PRA, permettendo il raffinamento delle previsioni di rischio associate agli eventi che riguardano il sistema (Figura 1.11)), (Compagno & al., 2008).

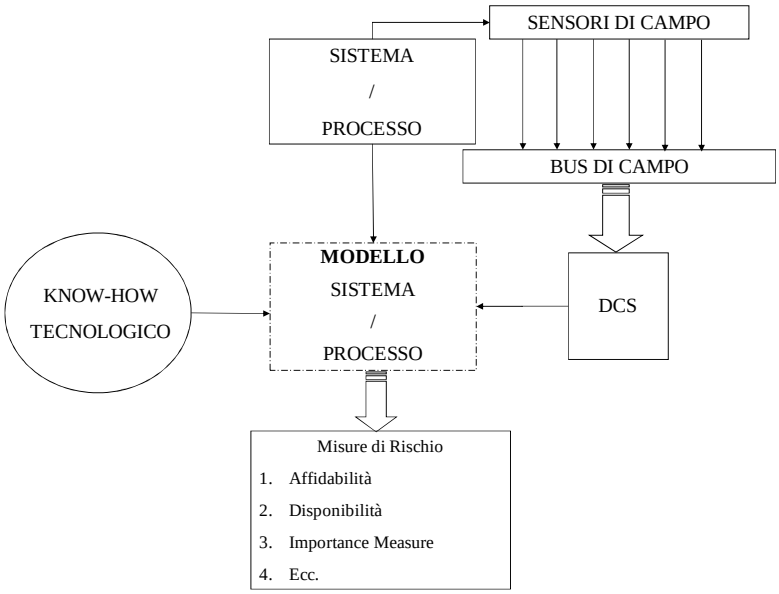


Figura 1.11: visione sistemica del PRA. Attorno al modello sistema/processo è osservabile il loop su cui si la diagnostica in tempo reale per la corretta valutazione delle misure di rischio

Il contributo che si vuole apportare è di ampliare la consapevolezza nell'uso dei DFT, sia in ambito affidabilistico sia in quello manutentivo. Infatti, la visione classica dei DFT come DAG (Direct Acyclic Diagram) (Merle et al., 2010), può essere ampliata in modo da arricchire l'autorevolezza della PRA con risultati che possano essere utilizzati per valutazioni riguardanti la disponibilità e l'orizzonte manutentivo.

2. LA MODELLAZIONE AFFIDABILISTICA

In questo capitolo vengono introdotti alcuni fra gli strumenti di modellazione più utilizzati nella PRA.
I modelli stocastici basati sulla logica combinatoriale, gli RBD e gli SFT, vengono richiamati sulla base della nozione della formula di struttura dei sistemi coerenti che sarà data nel paragrafo seguente.
In seguito, richiamando la teoria di Markov, si passa alla descrizione dei modelli nello spazio degli stati e dei DFT.
Viene poi trattata la problematica della risoluzione dei modelli complessi attraverso le tecniche dinamiche, mostrando le potenzialità aggiuntive di queste rispetto alle metodologie di analisi tradizionali (statiche).
In seguito viene richiamata la tecnica di decomposizione introducendo il concetto della gerarchizzazione esatta (strong) e non esatta (weak).
Conclude il capitolo una rivisitazione delle misure di importanza e della sensitività applicata ai modelli dinamici.

2.1 FORMULA DI STRUTTURA

Si consideri un sistema costituito da *n* componenti distinti, allora sarà possibile definire il vettore degli elementi $\underline{C} = \{c_1, c_2, \dots, c_n\}$, dove ogni elemento c_i rappresenta lo stato dell'i-esimo componente, ossia è tale che:

$$c_i = \begin{cases} 1, se\ il\ componente\ funziona \\ 0, se\ il\ \ componente\ \text{è}\ guasto \end{cases}.$$

Nell'ambito della dependability e della modellazione di un sistema binario, viene definita formula di struttura (Birnbaum, 1969), (Ebrahimi, 1990), (Fricks & Trivedi, 2003), l'espressione algebrica $\phi: \{0,1\}^n \rightarrow \{0,1\}$, dove (analogamente a quanto

stabilito per i componenti) si assume il valore 1 quando il sistema è funzionante e 0 se il sistema è guasto.

La formula di struttura è usata per stabilire una relazione tra lo stato di ogni componente e quello del sistema e permette l'analisi di modelli combinatoriali basando la risoluzione sulle regole dell'algebra booleana.

All'interno della popolazione $\underline{C}$ è possibile individuare (Zang et al., 2002), (Aven & Jensen, 1998), (Locks, 1978):

1. *k* sottoinsiemi, detti *path set*, $\underline{C}_1 = \{c_i : \phi(\underline{C}) = 1\}$ dati dalle combinazioni di quegli elementi c_i del sistema che, se funzionanti ($c_i = 1$), garantiscono il funzionamento del sistema. Un path è minimo (*minpath*) se non può essere ridotto, da cui cioè non è possibile rimuovere componenti senza che esso continui ad essere un path.
2. *j* sottoinsiemi, detti *cut set*, $\underline{C}_0 = \{c_i : \phi(\underline{C}) = 0\}$ dati dalle combinazioni di quegli elementi c_i del sistema che, se guasti ($c_i = 0$), garantiscono il non funzionamento del sistema. Un cut set è minimo (*mincut*) se non può essere ridotto, da cui cioè non è possibile rimuovere componenti senza che esso continui ad essere un cut set.

Definizione 2.1: Un sistema è monotono se:

a. $\boldsymbol{\phi(\underline{C})}$ è monotona non decrescente in ogni suo argomento, cioè per ogni c_i si ha che $\boldsymbol{\phi(c_1, c_2, \dots, c_{i-1}, 1, c_{i+1}, \dots, c_n) \geq \phi(c_1, c_2, \dots, c_{i-1}, 0, c_{i+1}, \dots, c_n), i = 1, \dots, n}$. In termini affidabilistici questa condizione implica che la rottura di un componente può solo diminuire l'affidabilità del sistema o, in altre parole, che aumentando le prestazioni affidabilistiche di un componente un sistema non può deteriorarsi;
b. $\phi(0) = 0$ *e* $\phi(1) = 1$, cioè se tutti i componenti del sistema sono in uno stato di guasto, allora il sistema è sicuramente guasto; viceversa se tutti i componenti sono funzionanti, il sistema è in uno stato di funzionamento.

Definizione 2.2: Un sistema è **coerente** se valgono le seguenti condizioni:

a. la sua funzione di struttura è monotona non decrescente in ogni suo argomento;
b. tutti i componenti del vettore $\underline{C}$ sono rilevanti, cioè, per ogni componente c_i esiste almeno un vettore $\underline{C}$ tale che $\phi(1_{i,}c) = 1$ e $\phi(0_{i,}c) = 0$. In pratica questo significa

che non esiste alcun componente c_i che non influenza il comportamento affidabilistico del sistema.

Si osserva che se $\boldsymbol{\phi}$ è coerente allora è anche monotona.

L'affidabilità di un sistema può dunque essere valutata mediante la funzione di struttura. Infatti, si ha:

$$R_{sys}(t) = P(\phi(\underline{C}) = 1) \qquad (E.2.1)$$

2.2 MODELLAZIONE STATICA

La modellazione statica è un comodo formalismo utilizzato nelle valutazioni di rischio tradizionali laddove non vengono considerate le dipendenze fra gli eventi e per cui la cronologia con cui possono avvenire i guasti dei componenti non ha alcuna rilevanza sul comportamento del sistema. Questo tipo di modellazione ha avuto molta diffusione successivamente agli anni '60 quando la materia della PRA iniziava ad assumere un ruolo centrale nella progettazione delle missioni aerospaziali (Rackley, 1976) e nelle valutazioni dei rischi per le centrali nucleari (NUREG-75/014, 1975), (IEEE, 1975). Secondo il formalismo statico, il calcolo dell'affidabilità di un sistema può essere ricondotto al calcolo della probabilità di tutte le combinazioni nello spazio campione $\underline{C}$, per cui la (*E.2.1*) è verificata: RBD e FT sono le due tecniche di modellazione stocastiche più famose per le valutazioni quantitative.

2.3 RELIABILITY BLOCK DIAGRAM (RBD)

I RBD sono uno strumento di modellazione molto potente nell'ambito della disciplina dell'affidabilità. Essi appartengono alla classe dei modelli stocastici combinatoriali (Murphy & Carter, 2003), (Sahinoglu et al., 2004).

La loro rappresentazione è molto intuitiva per cui la costruzione di un diagramma a blocchi risulta di facile implementazione e analisi. Infatti, se un RBD non è troppo esteso, la semplice analisi visiva porta a comprendere quali siano le parti del sistema necessarie al funzionamento del sistema.

Nella sua forma più semplice, i blocchi che costituiscono un RBD sono connessi fisicamente in serie e/o in parallelo. Due o più blocchi sono in serie quando il guasto di uno solo dei blocchi provoca il guasto di tutto il sistema, viceversa sono in parallelo quando il guasto del sistema si verifica solo in seguito al guasto di tutti i blocchi.

I RBD prevedono la costruzione di architetture più complesse mediante la combinazione di blocchi serie/parallelo o attraverso inserzioni di blocchi mediante connessione a ponte. Quando queste forme non possono essere ricondotte alle strutture serie/parallelo allora il sistema ha una topologia complessa (Murphy & Carter, 2003) e non è risolvibile con i metodi tradizionali dei RBD.

L'ordine di posizione dei blocchi in serie può essere arbitraria (Birolini, 2003). Il reticolo che ne risulta astrae il comportamento affidabilistico del sistema.

Nel formalismo degli RBD, un sistema è affidabile se esiste almeno un percorso tra l'ingresso e l'uscita del RBD. Quando un blocco è considerato fuori uso, si immagina una disconnessione fisica agli estremi del componente in questione.

Per la risoluzione di un RBD si fanno le seguenti ipotesi:

1. il comportamento affidabilistico di ogni singolo blocco non dipende da alcun altro blocco;
2. all'istante iniziale di osservazione (generalmente settato al tempo $t = 0$) tutti i blocchi sono completamente funzionanti;
3. per $t \geq 0$, ogni blocco può essere esclusivamente in uno stato di funzionamento o di non funzionamento;
4. gli elementi possono comparire più di una volta nel RBD;
5. sono ammesse solo ridondanze attive (i blocchi in parallelo sono tutti attivi a partire da $t \geq 0$).

Queste ipotesi permettono la risoluzione degli RBD mediante l'algebra booleana (Birolini, 2003); essa è usata per descrivere la combinazione minima di elementi il cui guasto causa un'interruzione del percorso tra monte e valle.

Definizione 2.3: Un RBD è completamente definito dalla quadrupla (C, L, N, J), dove:

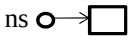
1. 'C' indica l'insieme dei componenti o blocchi;

2. 'L' indica l'insieme delle connessioni che possono insistere fra una coppia di blocchi dell'insieme C;

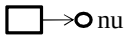
3. 'N' è l'insieme dei nodi. Deve sempre esistere un nodo *sorgente (ns)* ed un nodo di *uscita (nu).*

Valgono inoltre le seguenti relazioni 'J' di connessione:

i. N x L x C, rispetto al nodo iniziale e si traduce in almeno una relazione del tipo:

ns x L x C. 

ii. C x L x N, rispetto al nodo finale e si traduce in almeno una relazione del tipo:

C x L x nu. 

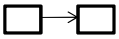
iii. C x L x C, che indica le interconnessione interne al RBD. 

Figura 2.1: Rappresentazione della terna (C, L, N) degli elementi che definiscono un RBD

I blocchi rappresentano delle entità fisiche; tuttavia possono anche rappresentare le fasi di un processo o di un'attività.

La Figura 2.2 mostra un semplice RBD definito mediante i seguenti insiemi:

1. C = {c1, c2, c3};
2. L = {l1, l2, l3, l4, l5};
3. N = {ns,nu};
4. le relazioni J={ns x l1 x c1; c1 x l2 x c2; c1 x l3 x c3; c2 x l4 x nu; c3 x l5 x nu}.

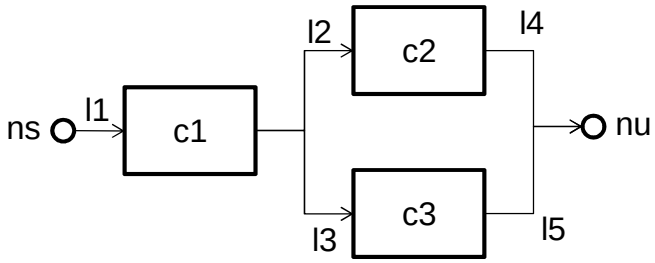


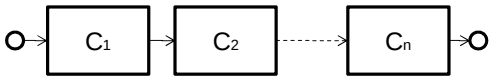
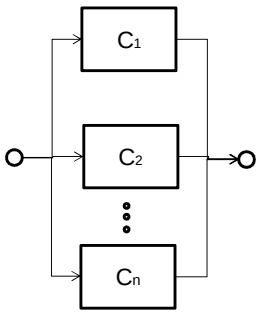
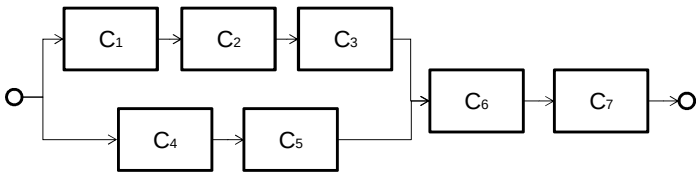
Figura 2.2: Sistema RBD formato da un parallelo di due componenti in serie con un altro blocco

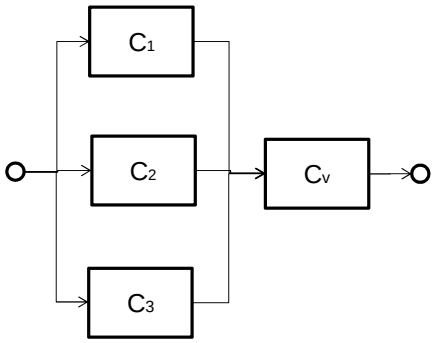
La formalizzazione di un RBD, come presentata nella Definizione 2.3, diventa utile nella costruzione dei metadati che si utilizzano per la codifica di routine informatiche di risoluzione.

La risoluzione di un RBD è basata sulla determinazione dei percorsi tra i nodi ns e nu: l'affidabilità R del sistema è data dall'unione delle probabilità delle configurazioni funzionanti, cioè di tutti i percorsi attivi tra ns ed nu.

In Tabella 2.1 sono riportate le formule per il calcolo dell'affidabilità di alcune topologie di RBD comuni, risolvibili mediante tecniche booleane.

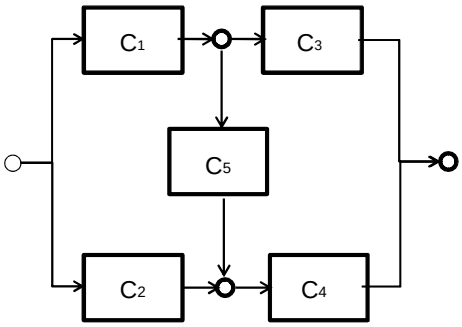
Tabella 2.1: calcolo dell'affidabilità per alcune configurazioni notevoli di RBD

Reliability Block Diagram	**Funzione di Affidabilità** ($R_s=R_s(t)$; $R_i=R_i(t)$; $R_i(0) = 1$)
	$R_s = R_i$
	$R_s = \prod_{i=1}^{n} R_i$
	$C_1 = C_2 = C_n = C$, $R_1 = R_2 = R_n = R$, $R_s = \sum_{i=1}^{n} \binom{n}{i} R^i (1-R)^{n-i}$
	$R_s = (R_1R_2R_3 + R_4R_5 - R_1R_2R_3R_4R_5)R_6R_7$

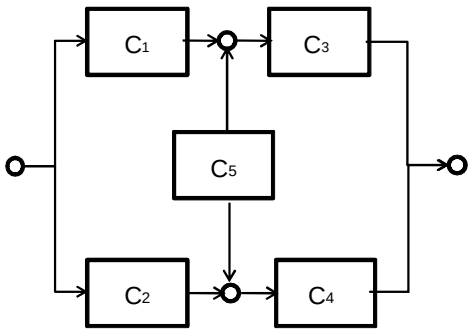


$C_1 = C_2 = C_3 = C$, $R_1 = R_2 = R_3 = R$,

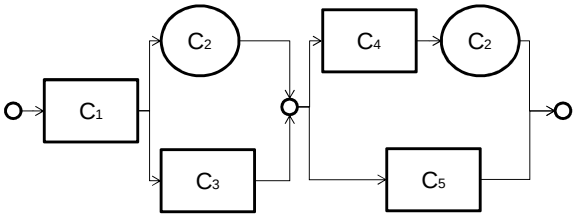
$$R_s = (3R^2 - 2R^3)R_v$$



$$R_s = R_4[R_2 + R_1(R_3 + R_5 - R_3R_5) - R_1R_2(R_3 + R_5 - R_3R_5)] + (1 - R_4)R_1R_3$$



$$R_s = R_5(R_1 + R_2 - R_1R_2)(R_3 + R_4 - R_3R_4) + (1 - R_5)(R_1R_3 + R_2R_4 - R_1R_2R_3R_4)$$



$$R_s = R_2R_1(R_4 + R_5 - R_4R_5) + (1 - R_2)\,R_1R_3R_5$$

2.4 FAULT TREE ANALYSIS

La metodologia del Fault Tree Analysis (FT-A) fu sviluppata per la prima volta nel 1962, all'interno dei laboratori della Bell Telephone per la US Air Force negli studi di affidabilità del sistema Minuteman (Watson, 1962).La FT-A è una tecnica di tipo top-

down di natura deduttiva (Vesely et al., 1981) che permette la rappresentazione di un sistema fisico in un diagramma logico strutturato (il Fault Tree) per cui certe specifiche cause pilotano il sistema verso un evento di particolare interesse, il Top Event (TE) (Lee et al., 1985). Nello studio del Fault Tree (FT), gli esperti del sistema fissano a priori quali sono gli eventi cruciali per la normale operatività del sistema/componente (i TE del sistema) e, attraverso un'analisi dall'alto verso il basso (dagli effetti si risale alle possibili cause), cercano di valutare quali siano le motivazioni che contribuiscono al verificarsi di detti TE. Quindi, mentre il TE e gli eventi intermedi all'albero sono il risultato di ben precise combinazioni i cui meccanismi sono descritti mediante delle logiche formalmente definite, gli eventi iniziatori o primari (PE) o nodi N (Eventi Base - BE, Eventi Esterni - EE, Eventi Non Sviluppati - UE) non richiedono ulteriori indagini e vengono associati ai nodi del FT, costituendo gli input del modello. Generalmente possono essere rappresentati da guasti di componenti, da eventi di tipo naturale o da errori umani.

Sebbene l'analisi top-down fornisce anche una visione di insieme che mostra le circostanze che conducono al guasto finale, la finalità dell’analisi non è quella di individuare i guasti iniziali quanto le loro relazioni funzionali. È, infatti, tramite tali relazioni che si sviluppa la costruzione dell'albero.

Nelle applicazioni reali, esempi classici di TE possono essere:

- il guasto di veicoli di una certa criticità (come l'esplosione in volo di un velivolo o il guasto di un aereo che richiede un atterraggio di emergenza) tali per cui si possa distinguere tra "guasti causati da eventi esterni" o "guasti dovuti alle normali operazioni di funzionamento";
- il malfunzionamento dei dispositivi di sicurezza in impianti a rischio rilevante (raffinerie, centrali nucleari) che causa incendi, esplosioni o reazioni irreversibili;
- il black-out o il malfunzionamento di una rete di telecomunicazioni o di una rete di distribuzione energetica dovuta al sovraccarico del traffico, alla disconnessione di nodi di smistamento (hub) o ad attacchi informatici.

Dal punto di vista della modellazione, l'analisi di un FT viene effettuata tenendo in considerazione il contesto di funzionamento del sistema, al fine di determinare tutte le potenziali modalità secondo cui il TE avviene.

Il FT in sé è un modello grafico delle varie combinazioni dei PE che conducono al verificarsi del TE. I PE possono essere associati a guasti hardware, ad errori umani o a qualsiasi altro tipo di evento collegato al cattivo funzionamento del sistema. Il FT, dunque, raffigura le interrelazioni logiche dei PE che conducono all'evento indesiderato.

Un FT non è un modello di tutti i possibili guasti del sistema; di conseguenza per ogni scenario individuato viene costruito un modello FT diverso. Inoltre, spesso, i PE considerati non sono esaustivi della dinamica del processo di guasto ma riguardano solo gli eventi più credibili indicati dagli esperti. Infatti, l'analisi qualitativa che conduce alla definizione degli scenari incidentali è molto soggettiva e per questa ragione è importante poter disporre di analisti del rischio che conoscano perfettamente il processo esaminato.

I simboli utilizzati per la costruzione grafica del diagramma logico ad albero FT, sono chiamati porte logiche e sono simili ai simboli utilizzati dai progettisti di circuiti elettronici digitali (Figura 3.3).

I vantaggi di tale tecnica risiedono nel fatto che:

- un FT chiarifica e semplifica le logiche del sistema e le cause dell’accadimento del TE;
- la FT-A è una metodologia che consente l’identificazione dei punti deboli del sistema mediante la valutazione di classificazione e sensitività degli elementi del sistema;
- la FT-A può essere utilizzata anche come design tool per la valutazione e il confronto dei possibili scenari di rischio, aiutando nella scelta della migliore configurazione che soddisfa i requisiti di design.

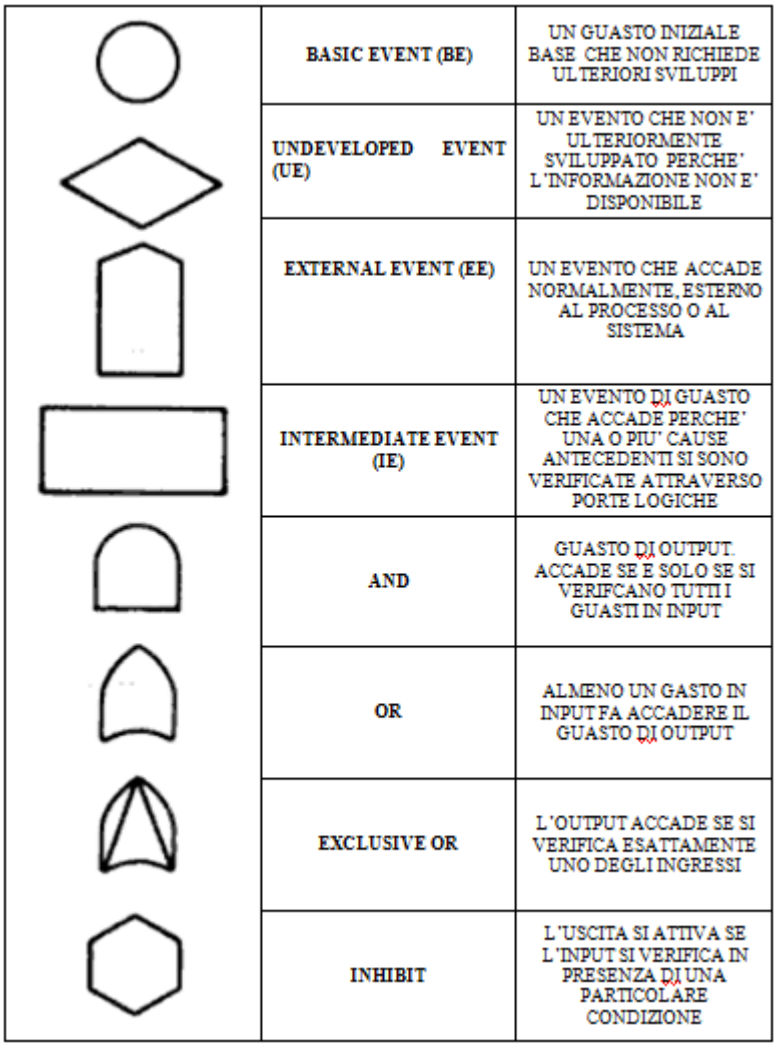
	**BASIC EVENT (BE)**	UN GUASTO INIZIALE BASE CHE NON RICHIEDE ULTERIORI SVILUPPI
	UNDEVELOPED EVENT (UE)	UN EVENTO CHE NON E' ULTERIORMENTE SVILUPPATO PERCHE' L'INFORMAZIONE NON E' DISPONIBILE
	EXTERNAL EVENT (EE)	UN EVENTO CHE ACCADE NORMALMENTE, ESTERNO AL PROCESSO O AL SISTEMA
	INTERMEDIATE EVENT (IE)	UN EVENTO DI GUASTO CHE ACCADE PERCHE' UNA O PIU' CAUSE ANTECEDENTI SI SONO VERIFICATE ATTRAVERSO PORTE LOGICHE
	AND	GUASTO DI OUTPUT ACCADE SE E SOLO SE SI VERIFCANO TUTTI I GUASTI IN INPUT
	OR	ALMENO UN GASTO IN INPUT FA ACCADERE IL GUASTO DI OUTPUT
	EXCLUSIVE OR	L'OUTPUT ACCADE SE SI VERIFICA ESATTAMENTE UNO DEGLI INGRESSI
	INHIBIT	L'USCITA SI ATTIVA SE L'INPUT SI VERIFICA IN PRESENZA DI UNA PARTICOLARE CONDIZIONE

Figura 2.3: Simbologia grafica di un fault tree

2.4.1 PROGETTAZIONE DI UN FAULT TREE

La progettazione di un albero di guasto (Figura 3.4) è generalmente un'attività complicata e onerosa in termini di tempo poiché presuppone l'esatta conoscenza dei processi funzionali e dei requisiti del sistema. Per definire chiaramente il sistema devono essere considerate tutte le funzioni in esso coinvolte e le loro connessioni. Inoltre si deve avere conoscenza dei target di performance e delle tolleranze di ciascuna delle funzioni. Per raggiungere un simile risultato sono necessari documenti tecnici, specifiche di performance e disegni di progetto. Per illustrare, poi, le

connessioni di sistema e le influenze di interfaccia, sono impiegati i diagrammi funzionali a blocchi.

Il sistema si trova a dover rispettare e mantenere i requisiti funzionali sotto l'influenza di condizioni ambientali specifiche non strettamente attinenti gli aspetti tecnici del sistema, durante diverse fasi operative. Al pari delle caratteristiche fisiche e chimiche degli elementi del sistema, devono essere considerate anche le influenze ambientali. La significatività di una FT dipende dalla descrizione del TE e dalle relative condizioni di frontiera. Nella determinazione del TE due sono i possibili approcci:

- preventivo, se la FT-A viene condotta a scopi preventivi; le definizioni dei TE scaturiscono da non conformità di funzioni o dalla necessità di soddisfare specifici requisiti. Nella definizione del TE, inoltre, al pari delle conformità del prodotto andranno ugualmente considerati aspetti legati alla sicurezza;
- correttivo, se il TE viene definito sulla base di un problema insorto o di un guasto al sistema già verificatosi.

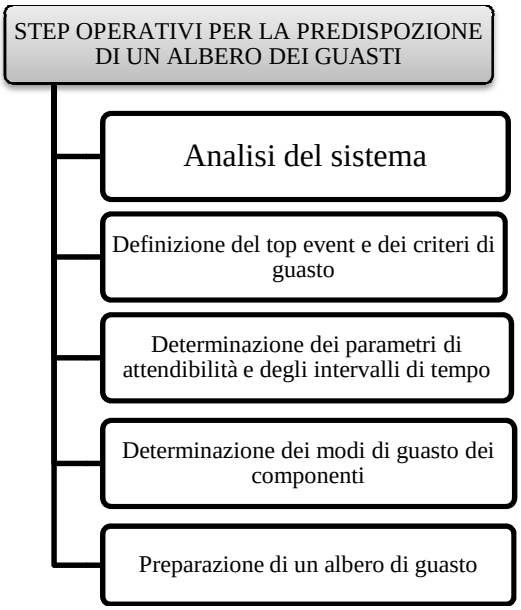


Figura 2.4: Step operativi per la predisposizione di un fault tree

Nel valutare quantitativamente un albero dei guasti, occorrerà distinguere i casi in cui si intende valutare le probabilità di guasto in un determinato periodo di tempo da quelli in cui questa stessa vada considerata per periodi casuali.

Dopo un'analisi del sistema e la definizione del TE, tutti i modi di guasto dei componenti coinvolti nel modello dell'albero devono essere considerati. Per

l'elaborazione di un FT-A dettagliato, normalmente non è sufficiente usare guasti indifferenziati dei componenti dei PE. Al contrario, modi di guasto diversi di uno stesso componente possono avere effetti completamente diversi sul TE, così che questi possano essere associati ad un medesimo elemento PE e dover tuttavia essere inseriti in un'area diversa dell'albero, se non addirittura in un altro modello FT. Quando si è chiamati a determinare l'attendibilità dei parametri dei PE sorge una difficoltà aggiuntiva legata al fatto che le probabilità di guasto sono conosciute ma non sono distinte in funzione del loro specifico effetto sul TE per i singoli modi di guasto. Alcuni data book contengono informazioni sui modi di guasto di alcuni componenti. Se non sono disponibili informazioni quantitative sui modi di guasto, è possibile in ultima analisi riferirsi all'ipotesi peggiore usando la probabilità di guasto complessiva di un componente come stima della più elevata probabilità del singolo modo di guasto.

2.4.2 RISOLUZIONE E VALUTAZIONE QUANTITATIVA DI UN FAULT TREE

Esistono diversi algoritmi per la risoluzione di un albero di guasto. Da ora in poi definiremo statici (Static Fault Tree - SFT), gli alberi di guasto che possono essere risolti attraverso le tecniche basate sull'algebra booleana, analogamente a quanto visto per gli RBD.

Definizione 2.4: Un SFT è completamente definito mediante la quadrupla (TE, PE, G, R), dove:

1. 'TE' indica il TOP EVENT del fault tree; è un elemento univoco e deve sempre esistere. Il TE è l'output logico della porta di livello più alto, la G_{TE}.
2. 'PE' indica l'insieme degli eventi primari o nodi, dato dall'unione dei BE, UE, EE input di un fault tree;
3. 'G' è l'insieme delle porte logiche statiche. L'uscita $O(G_i)$ di tutte le porte $G_i \neq G_{TE}$, rappresenta sempre un evento intermedio, ingresso di una porta di livello più alto;

4. inoltre, definito l'insieme B = O(G) ∪ PE, valgono le seguenti relazioni 'R' di connessione:

i. TE x G_{TE} x B, rispetto al TE che si traduce in una ed una sola relazione del tipo: TE x G_{TE} x B_i, dove B_i è un sottoinsieme dell'insieme B;

ii. O(G) x G x B, rispetto ai livelli intermedi dell'albero e si traduce in 0 o più relazioni del tipo: $O(g_i)$ x g_i x B_j , dove B_j è un sottoinsieme dell'insieme B, mentre g_i è un elemento dell'insieme G;

iii. O(G) x G x PE, rispetto ai livelli più bassi dell'albero che si traduce in 0 o più relazioni del tipo: $O(g_i)$ x g_i x PE_j, dove PE_j un sottoinsieme di PE e g_i un elemento dell'insieme G.

Gli elementi di PE rappresentano delle entità fisiche oppure possono rappresentare concetti più astratti come le fasi di un processo o di un'attività; gli elementi O(G), invece, sono associati alle conseguenze di una dinamica combinata per cui spesso astraggono la rottura di un sistema articolato o l'esito di un processo rilevante del modello FT.

In Figura 2.5 è mostrato un semplice esempio di FT, definito dai seguenti elementi:

1. TE;
2. PE = {c1, c2};
3. G = {G_{TE}} con G_{TE} = AND1;
4. B = N e l'unica relazione R ={TE x G_{TE} x(c1,c2)} di tipo (*4.i*) che vale rispetto al TE.

Quest'esempio è utile a dimostrare che l'unica relazione necessaria per definire uno SFT è l'esistenza di almeno una porta, la G_{TE}, la cui uscita rappresenta il TE e che mette in relazione gli elementi dell'insieme degli eventi primari PE, input dello SFT.

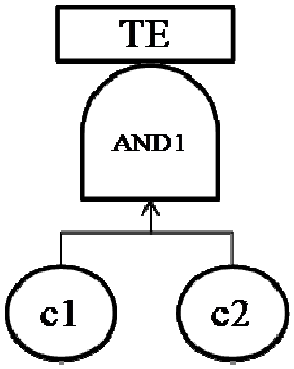


Figura 2.5: Esempio di FT ad un unico livello

Un altro esempio di FT, in Figura 2.6, viene definito mediante i seguenti elementi:

1. TE (se non è visibile in testa all'albero, si assume comunque come l'output della G_{TE});
2. N = {c1, c2, c3};
3. G = {G_{TE}, AND1}, con G_{TE}=OR1;
4. B = {N, O(AND1), c2, c3}, con le seguenti relazioni

   R = {

   (i) TE x G_{TE} x (c1,O(AND1));

   (ii) O(AND1) x AND1 x (c2,c3);

   };

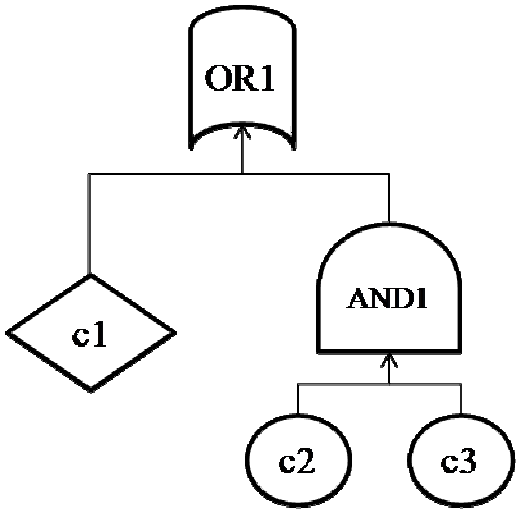


Figura 2.6: Altro esempio di FT

La formalizzazione di un SFT, come presentata nella Definizione 2.4, diventa utile nella costruzione dei metadati che si utilizzano per la codifica di routine informatiche di risoluzione.

La valutazione qualitativa di un SFT consiste nella determinazione dei Minimal Cut Set (MCS). Un MCS rappresenta l'insieme minimo degli eventi primari la cui simultanea occorrenza causa il TE. La valutazione deterministica dei MCS può rivelarsi un problema NP-completo. Per questo motivo esistono anche metodi basati sulla simulazione Monte Carlo.

L'idea alla base degli algoritmi deterministici è quella dell'espansione diretta e della successiva riduzione della formula del TE in termini di eventi primari, usando l'algebra booleana e pervenendo alla formula di struttura del FT. Per sistemi di grandi dimensioni, la determinazione dell'insieme degli MCS diventa molto complicata poiché i tempi di computazione e la richiesta di memoria diventa proibitiva. Per superare questi limiti, sono state proposte diverse procedure in grado di localizzare tutti i MCS di qualsiasi ordine che possono essere determinati da cause comuni (Wagner et al., 1978).

La simulazione Monte Carlo, invece, viene eseguita assegnando un tempo di guasto a tutti i componenti, generando un numero casuale uniformemente distribuito tra 0 e 1 e risalendo mediante inversione della formula della CDF al tempo di guasto corrispondente. Un'iterazione della simulazione prevede quindi di ordinare lo stato dei componenti uno per volta a seconda del loro tempo di guasto fin quando non si perviene al TE. Questa procedura genera un cut set che viene poi ridotto in un minimal cut set (Salem et al., 1978).

Analogamente a quanto visto per gli RBD, la valutazione quantitativa di un FT può essere effettuata se:

1. il comportamento affidabilistico di ogni singolo evento primario è indipendente dagli altri;
2. all'istante iniziale di osservazione (generalmente settato al tempo $t = 0$) tutti gli input del FT si presentano nel loro stato di corretta operatività;
3. per $t \geq 0$, ogni evento primario può essere esclusivamente in uno stato di funzionamento o meno e la transizione fra questi due stati è istantanea;

4. gli eventi primari possono comparire più di una volta nello SFT e, in questo caso, si parla di eventi ripetuti e di alberi Multi Occurrence Event (MOE).

La valutazione quantitativa di uno SFT parte dall'identificazione dei MCS. Valutare quantitativamente un albero di guasto significa calcolare la probabilità del TE, attraverso quella dei MCS. Gli MCS vengono determinati mediante le relazioni booleane che legano fra loro gli eventi primari che dipendono dalla struttura del sistema e dalle porte che lo compongono.
La porta AND propaga il guasto solo se tutti i componenti si guastano. Dati n input in ingresso, l'uscita della porta viene calcolata attraverso la seguente:

$$\boldsymbol{F_{AND}(t) = \prod_{i=1}^{n} F_i(t)} \qquad \textit{(E.2.2)}$$

La porta OR propaga il guasto se anche solo uno dei componenti si guasta:

$$\boldsymbol{F_{OR}(t) = \cup_{i=1}^{n} F_i(t)} \qquad \textit{(E.2.3)}$$

Il TE non è altro che l'unione dei MCS, dunque la probabilità di accadimento può essere calcolata come la probabilità dell'unione dei MCS:

$$P(TE) = P(\cup_i MCS_i) \qquad \textit{(E.2.4)}$$

Ogni MCS viene etichettato con un ordine che indica il numero di elementi dell'insieme (es: un MCS di ordine 3 contiene 3 eventi primari). I MCS più critici sono quelli di ordine più basso, poiché in linea di massima hanno una più alta probabilità di causare l'evento TE (a parità di probabilità di occorrenza di un evento primario) essendo più direttamente collegati con la G_{TE}. Un sistema che ha molti MCS di ordine basso indica poca robustezza della struttura.
In base all'ordine dei MCS possono essere valutate le possibilità di utilizzare l'approssimazione agli eventi rari (Brown, 1990), (Modarres et al., 1999) per cui, se $P(MCS_i) < (50n)^{-1}$, con n pari al numero di MCS individuati, si può scrivere:

$$\boldsymbol{P(TE) = P(\cup_i MCS_i) = \sum_i P(MCS_i)} \qquad \textit{(E.2.5)}$$

Esistono svariati metodi per la ricerca efficiente dei MCS. In (Bennets, 1975) viene dimostrata l'efficienza di un algoritmo per il calcolo dei *sop* (sum of products) degli eventi primari, mentre in (Rauzy, 1993), (Bryant, 1992), (Sinnamon & Andrews, 1996) si valida la bontà del calcolo esatto attraverso l'uso degli alberi binari di decisione (BDD) che risultano di estrema efficacia anche in caso di alberi MOE. In successivi lavori quali (Limnios & Ziani, 1986), (Odeh & Limnios, 1994) vengono sviluppate tecniche per velocizzare le performance di calcolo e di valutazione degli MCS, tutte indirizzate alla codifica di routine informatiche.

2.5 RBD E FT A CONFRONTO

Un modello RBD ammette sempre una rappresentazione duale in termini di FT. Mentre il primo è un modello che sintetizza l'insieme delle configurazioni funzionanti di un sistema e ne calcola la probabilità pervenendo dunque all'affidabilità, il FT è un modello che ne rappresenta le configurazioni di guasto pervenendo, quindi, all'inaffidabilità del sistema.

Quando si converte un RBD nel suo FT equivalente, viene effettuata una trasformazione tale da realizzare il completamento ad uno della relazione tra affidabilità (degli RBD) e inaffidabilità (del FT). Da un punto di vista topologico, sia gli RBD che i FT sono risolvibili come reti di Bayes, dal momento che entrambi sono per natura equivalenti a un DAG (Torres Toledano & Succar, 1998).

Per esempio, si consideri il sistema RBD di n componenti in serie di Figura 2.7.

L'affidabilità del sistema è: $R = \prod_{i=1}^{n} \boldsymbol{R_i}$. Passando all'inaffidabilità, tramite il complemento ad uno, si può scrivere $F = \mathbf{1} - \prod_{i=1}^{n} \boldsymbol{R_i} = \mathbf{1} - \prod_{i=1}^{n} (\mathbf{1} - \boldsymbol{F_i})$.

Per la formula di espansione si ha:

$$\boldsymbol{P}(\cup_i \boldsymbol{A_i}) = \sum_{1 \le i \le m} \boldsymbol{P}(\boldsymbol{A_i}) - \sum_{1 \le i < j \le m} \boldsymbol{P}(\boldsymbol{A_i} \cap \boldsymbol{A_j}) + \sum_{1 \le i < j < k \le m} \boldsymbol{P}(\boldsymbol{A_i} \cap \boldsymbol{A_j} \cap \boldsymbol{A_k}) + \cdots + (-\mathbf{1})^{m-1} \boldsymbol{P}(\boldsymbol{A_1} \cap \boldsymbol{A_2} \cap \ldots \cap \boldsymbol{A_k})$$

(E.2.6)

Per l'ipotesi di indipendenza degli eventi $\prod_{i=1}^{n}(\mathbf{1}-\boldsymbol{F_i}) = \mathbf{1} - \cup_{i=1}^{n}\boldsymbol{F_i}$ per cui l'ultima relazione può essere riscritta come segue: F=$\mathbf{1} - \mathbf{1} + \cup_{i=1}^{n}\boldsymbol{F_i} = \cup_{i=1}^{n}\boldsymbol{F_i}$, che corrisponde ad un sistema formato da una porta OR a n ingressi di un FT.

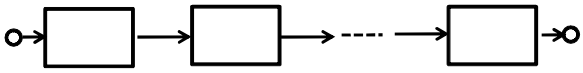


Figura 2.7:Sistema RBD in serie

Mediante ragionamenti analoghi a quelli precedenti è possibile concludere che un sistema parallelo di n componenti di un RBD corrisponde ad una porta AND ad n ingressi di un FT.

Anche il RBD di Figura 2.2 può facilmente essere convertito in un FT, partendo dal nodo di uscita del sistema. Il parallelo (C2, C3) viene sostituito con una AND (C1,C2), mentre la serie tra C1 e (C2, C3) da una OR. In questo modo si perviene all'architettura del FT in Figura 2.6.

Per sistemi molto estesi quest'operazione non è immediata per cui spesso bisogna ricorrere a tecniche di fattorizzazione, spaccando l'RBD in RBD più piccoli che, eventualmente, possono essere descritti da altrettanti FT equivalenti.

La scelta sull'uso di una delle due tecniche può dipendere dal tipo di applicazione sotto analisi e dai successivi usi riguardanti la gestione delle politiche di ottimizzazione e di manutenzione in termini di PRA.

Per esempio, i FT si prestano bene allo studio dei modi di guasto ed ai vincoli di operatività di un sistema e divengono particolarmente convenienti nella descrizione degli scenari di rischio, piuttosto che alla rottura di un sistema. Il cattivo funzionamento di un componente può essere descritto mediante un evento di guasto, ovvero di degradazione delle funzionalità; l'errore umano è meglio descritto da un BE di un FT che non da un blocco fisico di un RBD. Certamente la rappresentazione sistemica di un FT è meno intuitiva da interpretare rispetto a quella di un RBD che per natura, invece, si presta alla rappresentazione affidabilistica di un sistema. Gli RBD hanno altre interessanti qualità. Per esempio, i blocchi possono essere ordinati in funzione di esigenze manutentive oppure nella sequenza in cui le funzioni di un processo si susseguono. Questa modellazione può essere preferita per la descrizione

di processi come flussi di sistema, tubazioni idrauliche o elettriche (Gough et al., 1990).

2.6 SISTEMI DIPENDENTI: AFFIDABILITÀ, DISPONIBILITÀ E CALCOLO DELLA PROBABILITÀ DI OCCORRENZA DI UN EVENTO

La teoria matematica dell'affidabilità (Birolini, 2003), (Modarres et al., 1999) fornisce delle definizioni rigorose per le misure di affidabilità e disponibilità (UNI, 1991). Sebbene i modelli stocastici quantitativi tradizionali sono sviluppati attorno a queste definizioni, essi propongono un'impostazione più pratica al calcolo di queste misure con il risultato che, spesso, il modello del sistema o del processo va impostato tenendo conto di ulteriori ipotesi al contorno. Per esempio, negli SFT la riparazione di un componente riparabile appartenente all'insieme degli MCS di un sistema riparabile, comporta l'immediato ritorno in servizio del sistema. Quest'ipotesi rende immediato il calcolo della disponibilità di un sistema statico mediante le già note relazioni dell'algebra booleana. In realtà, tra i limiti evidenziati dalle tecniche di modellazione combinatoriali (Sharma & Bazovsky, 1993), (Dugan et al., 1992) vi è l'impossibilità di descrivere dei comportamenti chiave della coppia sistema/processo. L'esigenza di far fronte in maniera significativa a queste problematiche ha spinto ricercatori e analisti a cercare nuove soluzioni.

Nella pratica (soprattutto nello studio di sistemi complessi), in talune valutazioni di rischio, è possibile se non necessario individuare casi per cui abbia senso calcolare la probabilità di occorrenza di un evento, cioè la probabilità che un evento si manifesti. Questa misura non è altro che la probabilità di TE (nei FT) di un sistema costituito da almeno un componente riparabile ed spesso associata all'inaffidabilità del sistema secondo lo scenario descritto. Ma, per definizione, l'inaffidabilità è il calcolo della probabilità di un sistema di compiere correttamente e senza interruzione le funzioni assegnate. Per questo motivo, la reversibilità del sistema a TE occorso non è da includere nelle valutazioni di probabilità. Purtroppo, con le tecniche statiche come gli SFT e gli RBD la risoluzione di un modello che contiene anche un solo componente riparabile comporta automaticamente il calcolo della disponibilità del sistema poiché

per ipotesi di SFT, il sistema torna funzionante qualora un componente dei MCS viene riparato.

A causa di quanto detto, il calcolo dell'affidabilità di sistema o il calcolo della probabilità di occorrenza di un evento per sistemi costituiti da componenti riparabili non può essere valutato con le tecniche statiche e rientra nel ventaglio delle casistiche tipiche dei modelli complessi.

Per spiegare meglio quanto detto, si consideri il sistema AND a 2 ingressi di Figura 2.8 e la sua rappresentazione nello spazio degli stati per (a) l'affidabilità e (b) la disponibilità.

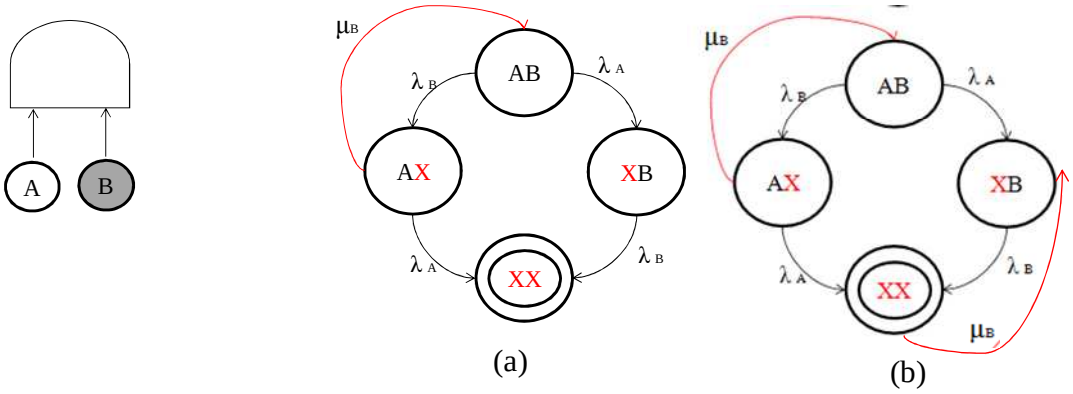


Figura 2.8: Affidabilità di una porta AND con un componente non riparabile (A) ed uno riparabile (B). Il formalismo grafico che si utilizza è che gli stati non affidabili sono quelli con il doppio cerchio.

Inoltre, si supponga che B sia un componente riparabile.

Il sistema può trovarsi in quattro stati diversi:

- 'AB' = entrambi i componenti funzionano e il sistema è funzionante;
- 'XB' = il componente A è guasto, B funziona quindi il sistema è funzionante;
- 'AX' = il componente A funziona, B è guasto quindi il sistema è funzionante;
- 'XX' = entrambi i componenti sono guasti quindi il sistema è guasto.

Da un punto di vista dell'affidabilità, fintanto che il sistema non transita per lo stato finale 'XX' (assorbente), se si è nello stato 'AX' è possibile ripristinare il sistema allo stato iniziale poiché B è riparabile. Questo tipo di valutazione non è deducibile mediante i modelli combinatoriali perché il calcolo attraverso quest'ultimi con la

l'algebra booleana realizza la logica (b), secondo cui la riparazione del componente B (che appartiene all'insieme degli MCS) riporterebbe il sistema ad uno stato di funzionamento ('XB').

Un caso pratico di questo problema viene sollevato in (Powers & Tompkins, 1974) nella valutazione del rischio di un reattore chimico, dove il TE è dato dall'esplosione del reattore stesso. In questo caso, prima che un'esplosione possa verificarsi devono accadere delle condizioni alcune delle quali sono reversibili; soltanto in seguito ad altri eventi il ripristino delle condizioni di sicurezza non è più possibile. Quindi, la dimensione temporale inizia a diventare essenziale per delle valutazioni realistiche di rischio. Per questo motivo, anche nell'ambito delle applicazioni industriali (in impianti a rischio di incidente rilevante), gli strumenti statici sono limitati poiché si richiede spesso il calcolo della probabilità che un evento indesiderato non accada mai.

Dunque, durante la fase di modellazione di uno scenario incidentale è importante chiedersi dall'inizio qual è l'obiettivo richiesto dall'analisi di rischio. Le domande che bisogna dunque porsi prima sono (Figura 2.9):

1. Il TE riguarda la valutazione della disponibilità del sistema o della sua affidabilità?
2. Gli input del modello (eventi base, eventi non sviluppati, eventi esterni) sono elementi che vengono sottoposti a possibile riparazione?
3. Se tali elementi sono riparabili, sono possibili riparazioni di questi elementi tali da riportare il sistema alle sue condizioni nominali di funzionamento?
4. Il TE è un evento irreversibile tale per cui è ha senso calcolare soltanto la probabilità della sua prima occorrenza?

L'ultimo quesito fornisce la giustificazione ai modelli di valutazione offerti dalla teoria di Markov poiché, grazie al modello nello spazio degli stati, diventa possibile la descrizione di dinamiche tempo dipendenti che sopperiscono alle insufficienze delle tecniche combinatoriali.

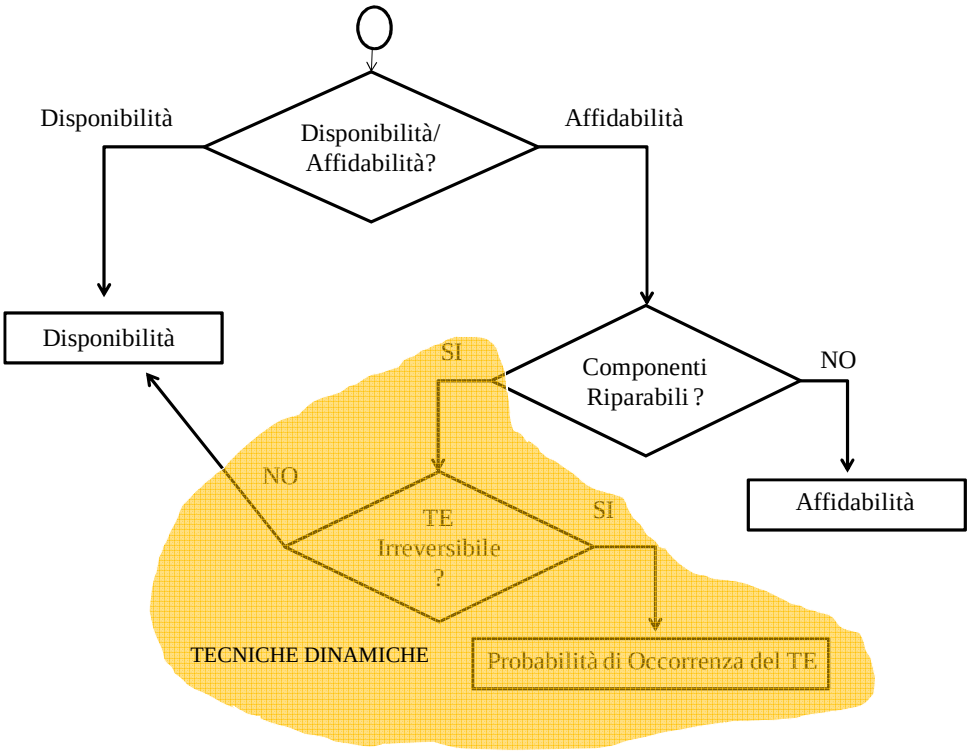


Figura 2.9: diagramma di flusso per la determinazione della misura di interesse

2.7 ANALISI DI SISTEMI DIPENDENTI ATTRAVERSO LA TEORIA DI MARKOV

I metodi tratti dalla teoria di Markov possono essere dei potenti tools nell'ambito RAMS, poiché forniscono il supporto necessario per analizzare sistemi che esibiscono delle dipendenze molto forti, superando il vincolo di indipendenza stocastica dei metodi combinatoriali (RAC START, 2003), (Sharma & Bazovsky, 1993).

Un sistema dipendente è costituito da un insieme di elementi ciascuno dei quali può essere in un dato istante funzionante, guasto o in standby (sono pensabili anche stati intermedi, quali per esempio quello di funzionamento parziale o degradato).

Per qualunque sistema valgono le seguenti convenzioni:

1. all'istante iniziale (t = 0) assumiamo che ciascun componente si trovi nello stato di corretto funzionamento (attivo, standby, ...);
2. in ogni istante ciascun componente funzionante può guastarsi;

3. in ogni istante ciascun componente guasto può essere riparato o sostituito e quindi ritorna ad essere funzionante;
4. il sistema passa da uno stato ad un altro ogni volta che un componente si guasta o viene riparato;
5. ogni combinazione degli stati dei singoli componenti rappresenta, in generale, uno stato differente per il sistema (a meno che non esistano configurazioni simmetriche);
6. gli eventi sono indipendenti l'uno dall'altro, ovvero il guasto di un componente non farà aumentare la probabilità di guasto di un altro componente (lo stesso vale per le riparazioni).

Nell'analisi di Markov, i tassi (nell'esempio λ e μ) vengono ipotizzati costanti. In queste condizioni il processo di Markov (D-C) è una Catena di Markov Tempo-Continua (CTMC) (Nielses, 2009).
Sebbene le CTMC siano molto potenti, esse possono trattare solo applicazioni che fanno uso della distribuzione esponenziale negativa per cui lo stato del sistema dipende soltanto dallo stato immediatamente antecedente. Questa ipotesi risulta dunque molto limitante. Una più efficacie rappresentazione nello spazio degli stati è offerta dalle MRGP e dalle GSMP (Sahner et al., 1996), (Kosiuczenko & Lajios, 2007) che tengono traccia della storia del sistema e permettono transizioni basate su distribuzioni generalizzate.
In Figura 2.10 è mostrata una semplice applicazione di sistema riparabile, in cui sono possibili solo 2 stati: X_1 rappresenta lo stato iniziale e di funzionamento per il sistema, X_2 di guasto. Il sistema può passare da uno stato ad un altro in qualsiasi istante.

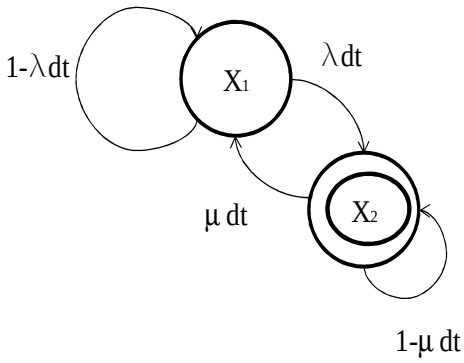


Figura 2.10: Rappresentazione nello spazio degli stati di un sistema riparabile

Per risolvere il modello si possono scrivere le equazioni di bilancio delle probabilità degli stati:

$$\begin{cases} p_1(t+dt) = p_1(t)(1-\lambda dt) + p_2(t)(\mu dt) \\ p_2(t+dt) = p_1(t)(\lambda dt) + p_2(t)(1-\mu dt) \end{cases}$$

La prima indica che la probabilità di trovarsi nello stato X_1 all'istante t+dt (che è lo stato infinitesimamente (dt) prossimo all'istante t) è pari alla probabilità di permanere nello stesso stato X_1 (quando all'istante t si è già in X_1) sommata alla probabilità di transitare dallo stato X_2 verso X_1 (se all'istante t si è in X_2). Questa condizione si esprime con la prima equazione del sistema. Analoghi ragionamenti si possono fare per lo stato X_2. Risolvendo il sistema di equazioni differenziali si ha:

$$\begin{cases} p_1(t+dt) - p_1(t) = [-\lambda\, p_1(t) + \mu\, p_2(t)]dt \\ p_2(t+dt) - p_2(t) = [\lambda\, p_1(t) - \mu\, p_2(t)]dt \end{cases}$$

e dividendo tutto per *dt:*

$$\begin{cases} \dfrac{p_1(t+dt) - p_1(t)}{dt} = [-\lambda\, p_1(t) + \mu\, p_2(t)] \\ \dfrac{p_2(t+dt) - p_2(t)}{dt} = [\lambda\, p_1(t) - \mu\, p_2(t)] \end{cases}$$

Passando al limite per $dt \to 0$ si ottiene il sistema di equazioni differenziali

$$\begin{cases} \dot{p}_1(t) = [-\lambda\, p_1(t) + \mu\, p_2(t)] \\ \dot{p}_2(t) = [\lambda\, p_1(t) - \mu\, p_2(t)] \end{cases}$$

che si può rappresentare anche in forma matriciale:

$$[\dot{p}_1(t) \quad \dot{p}_2(t)] = [p_1(t) \quad p_2(t)] \cdot \begin{bmatrix} -\lambda & \lambda \\ \mu & -\mu \end{bmatrix}$$

in forma compatta:

$$\dot{P}(t) = P(t)\,[Q_{ij}]$$

dove $[Q_{ij}]$ è detta **matrice dei tassi di transizione** (2×2) o dei generatori infinitesimali.

Si osserva che nella matrice Q dei tassi di transizione la somma degli elementi di una riga è sempre nulla, mentre nella matrice di transizione P la somma degli elementi di una riga è sempre pari a 1 (gli elementi della riga sono infatti probabilità):

$$\sum_{j=1}^{n} q_{ij} = 0 \qquad \sum_{j=1}^{n} p_{ij} = 1 \qquad \forall\ i = 1...n$$

Risolvendo il sistema di equazioni differenziali si ha:

$$p_1(t) = \frac{\mu}{\lambda+\mu} + \frac{\lambda}{\lambda+\mu} e^{-(\lambda+\mu)t}$$

La funzione $p_1(t)$ corrisponde alla disponibilità istantanea di un sistema riparabile, funzionante all'istante iniziale.

Se si calcola il limite per $t \to \infty$ di tale funzione, si ottiene la disponibilità di regime**:**

$$Q = Q(t \to \infty) = \frac{\mu}{\lambda+\mu} = \frac{\frac{1}{\lambda}}{\frac{1}{\mu}+\frac{1}{\lambda}} = \frac{MBTF}{MBTF + MTTR}$$

2.7.1 PROCEDURA STANDARD GENERALIZZATA

Si può generalizzare lo studio di un modello affidabilistico mediante i processi di Markov attraverso i seguenti passaggi:

1. si analizzano gli stati X_i e si disegna il diagramma degli stati, indicando vicino a ciascuna freccia uscente la probabilità di transizione verso lo stato X_j;
2. si scrive la matrice dei tassi di transizione Q ricordando che:
3. si sceglie il vettore probabilità iniziale P(0): solitamente lo stato iniziale ha probabilità 1;
4. si risolve il sistema di equazioni differenziali;
5. si identifica la funzione desiderata (R, A) come opportuna unione di alcune componenti del vettore P degli stati.

2.8 ANALISI DI SISTEMI DIPENDENTI ATTRAVERSO I DFT

Lo sviluppo della tecnica dei Dynamic Fault Tree (DFT) nasce dall'esigenza di unire la potenza e la flessibilità delle tecniche nello spazio degli stati con l'immediatezza delle rappresentazioni combinatoriali.

I DFT estendono gli SFT mediante l'aggiunta delle porte dinamiche, con caratteristiche tempo-dipendenti. Queste porte permettono la descrizione di sistemi con dinamiche e interazioni la cui complessità è notevolmente più elevata rispetto agli standard offerti dalle tecniche combinatoriali (Boudali et al., 2007).

Da un punto di vista formale, un SFT che contiene anche solo una porta dinamica diviene un DFT.

Fra un modello DFT e un processo di Markov esiste una relazione di isomorfismo per cui ad ogni rappresentazione ad alto livello (DFT) corrisponde una ben precisa rappresentazione nello spazio degli stati.

Il vantaggio che si ricava con questi strumenti è di spostare la difficoltà della modellazione a dei livelli più alti, mediante linguaggi più consoni alla modellazione di scenari complessi di rischio. Per gli analisti del rischio questi strumenti sono di grande interesse perché offrono una potente alternativa alle tecniche di Markov: da un lato essi conservano l'intuitività della rappresentazione sistemica delle tecniche combinatoriali e dall'altro permettono la descrizione di dinamiche e interazioni espresse con le tecniche nello spazio degli stati. Chiaramente, la difficoltà di risoluzione non è inferiore, per cui esistono software specifici che realizzano automaticamente la conversione e la successiva risoluzione dei modelli.

Le più importanti porte dinamiche (vedi Figura 2.11) introdotte sono la:

- PAND (Priority AND gate);
- FDEP (Functional Dependency gate);
- SEQ (Sequence Enforcing gate);
- SPARE (Spare gate).

Un'approfondita descrizione con esempi sull'uso delle porte dinamiche può essere rintracciata in (Dugan et al., 1992) e in (Xing & Amari, 2008).

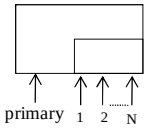
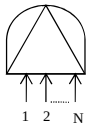
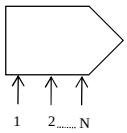
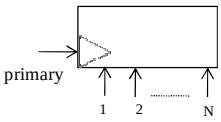
Nome	Rappresentazione Grafica	Descrizione (N ingressi)
SPARE		L'evento di output si verifica solo se accade l'evento primario (primary) e non sono più disponibili i dispositivi di ricambio (1,.., N) (che possono essere condivisi con altre porte spare).
PAND		Il funzionamento è identico alla AND sebbene l'evento si verifichi solo se gli ingressi accadono secondo la sequenza ordinata, da sinistra verso destra.
SEQ		Diversamente dalla PAND (per cui gli eventi possono accadere in un ordine casuale, determinando il tipo di output della porta) la SEQ forza gli aventi ad accadere secondo la sequenza ordinata 1,..,N. È ideale per descrivere i sistemi che si degradano.
FDEP		Questa porta viene utilizzata per esprimere la dipendenza di taluni eventi nei confronti di altri eventi (il primario). Quando il primario si verifica tutti gli eventi 1,…,N ad esso connessi vengono forzati ad accadere. L'uscita della FDEP è una dummy output.

Figura 2.11: Rappresentazione e descrizione sintetica delle più famose porte dinamiche

Il contenuto informativo delle porte dinamiche è completamente fornito dalla coppia (output, tempo) che caratterizza la porta. Entrambe queste informazioni vengono propagate risalendo l'albero poiché (nel caso in cui la porta in questione non è quella di TE) rappresentano l'ingresso ad altre porte.

La versatilità e la diffusione di queste porte è testimoniata dalla varietà di applicazioni che è succeduta alla loro ideazione come per esempio il calcolo della valutazione di sistemi informatici ad alta affidabilità (Dugan et al., 1992), (Xu & Dugan, 2004), del sistema avionico MAS (Gulati & Dugan, 1997), del sistema cardiaco assistito CAS (Boudali & Dugan, 2005), del multi-processore parallelo (Malhotra & Trivedi, 1995), ecc..

Infatti, qualitativamente le porte dinamiche permettono di modellare sistemi che presentano:

- diversi livelli di degradazione funzionale e, dunque, il calcolo dell'affidabilità parziale legata ad un particolare stato di un suo componente (mediante la SEQ);
- la rottura di un sistema associata all'ordine di rottura dei suoi componenti (PAND);
- la gestione delle ridondanze, cioè dei componenti di sostituzione (SPARE);
- la gestione dei meccanismi d'intervento per le logiche di controllo associate ai componenti di sostituzione (il cosiddetto 'coverage' factor) (SPARE-FDEP).

La modellazione mediante i DFT segue i passi riportati in Figura 2.12; come si può osservare da un confronto con gli SFT (Figura 2.4), la sequenza ricalca esattamente quella tipica della tecnica degli SFT: l'unica differenza in questi casi avviene nella fase di sintesi dell'albero dei guasti, potendo avvalersi della flessibilità delle porte dinamiche.

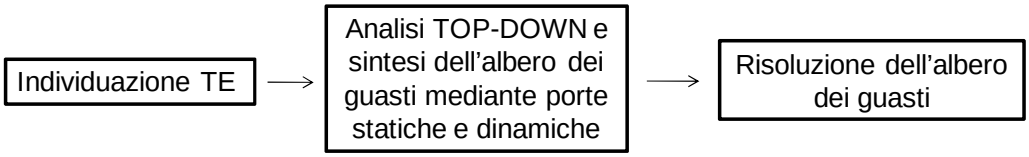


Figura 2.12: Passi per la risoluzione di un DFT

Come negli SFT, il formalismo sistemico di alto livello basato sulla costruzione di un albero in metodologia Top-Down è mantenuto. Questa caratteristica rappresenta un notevole punto di forza per gli addetti ai lavori. Infatti, la rappresentazione risulta di facile comprensione anche per i non esperti e permette un confronto costruttivo fra i diversi livelli aziendali.

2.8.1 RISOLUZIONE DEI DFT

Con i DFT, la risoluzione mediante l'algebra booleana è insufficiente poiché la formula di struttura è dipendente del tempo. A causa di ciò, le dipendenze implementabili con i DFT sono molto più numerose di quanto la logica binaria dell'algebra di Bool possa esprimere. Ciò che accade in termini sistemici è che delle interazioni fra le varie parti del sistema possono scaturire anche a causa di sequenze

temporali (assolutamente ignorate dalle tecniche combinatoriali). La dimensione temporale, quindi, arricchisce il numero di stati in cui un sistema può trovarsi.
Diversamente dagli SFT, nei DFT la dimensione temporale viene veicolata anche dalle porte dinamiche, poiché assumono un ruolo attivo per l'interazione dei PE.
Uno degli esempi più classici per la modellazione di una logica affidabilistica tempo-dipendente viene fornita da un sistema con un componente attivo ed uno in stand-by connesso mediante un interruttore controllato automaticamente. Se il commutatore si rompe dopo il componente primario allora il sistema può continuare a funzionare mediante il suo pezzo di ricambio in logica stand-by. Viceversa, nel caso in cui il commutatore si guasti prima del componente primario, la commutazione verso il componente in stand-by non è più automatizzata, comportando un arresto delle funzionalità osservate dal sistema (Dugan et al., 1992). La logica di guasto di questo sistema è dunque dipendente dalla sequenza con cui i guasti dei componenti avvengono. La modellazione nello spazio degli stati di questo sistema è descritta dalla Figura 2.13, dove in ordine si ha che P è il componente primario, c il commutatore e R il ricambio in configurazione stand-by freddo. Lo stato iniziale di partenza vede tutti i componenti del sistema funzionanti ed è tale che P garantisca il corretto funzionamento. La transizione verso lo stato "$\bar{P}cR$" indica il guasto del primario. In questo caso, lo stato descrive una commutazione automatica, tramite c, del sistema che rimane in uno stato di corretto funzionamento, mediante il componente di ricambio R. La transizione verso lo stato "$P\bar{c}R$" indica il guasto del commutatore automatico.

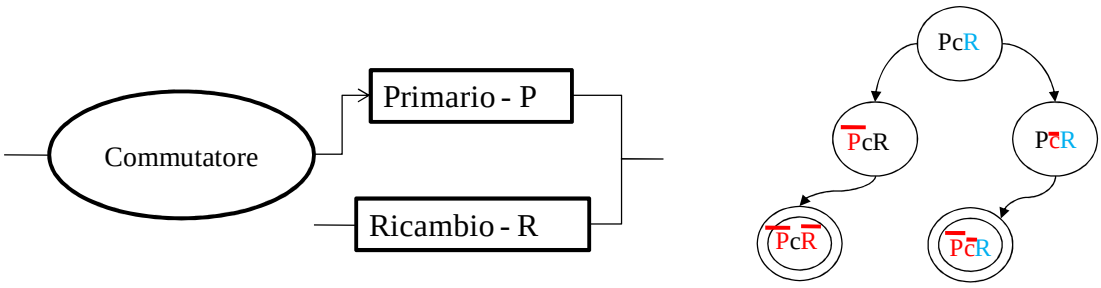


Figura 2.13: Sistema stand-by con commutazione automatica ed equivalente modello affidabilistico nello spazio degli stati

In questo stato il sistema funziona per mezzo del primario; una successiva rottura di tale componente porta il sistema al guasto poiché il ricambio non può essere abilitato dal controllore (stato "$\bar{P}\bar{c}R$").
La fase di risoluzione dell'albero dei guasti dinamico è l'aspetto più avvincente e critico della procedura di risoluzione, dal momento che non esistono tecniche ben definite.
La fattibilità dipende, infatti, da numerosi fattori:

- la struttura del DFT;
- il tipo di ingressi al modello;
- le misure che si vogliono ottenere.

In questo scenario, la capacità di realizzare un'analisi preliminare che permetta la comprensione del tipo di modello DFT può determinare sensibilmente la qualità dell'analisi effettuata, sia in termini di risultati ottenuti sia in termini di prestazione del risolutore.
Inoltre, nei DFT molto più che per un SFT, la risoluzione si lega agli strumenti di calcolo in forza agli analisti. Infatti, sono le routine informatiche che permettono l'automatizzazione di procedure per il processamento dell'albero e per i calcoli finali che altrimenti sarebbero inavvicinabili. In questa direzione si sono soffermati gli sforzi di molti ricercatori che hanno messo le basi per lo sviluppo di tool informatici esperti dediti alla risoluzione dei DFT mediante le tecniche di conversione nello spazio degli stati e della gerarchizzazione (Dugan et al., 1997), (Sullivan et al., 1999), (Dugan et al., 2000) come il Galileo, o il Relex® (Amari et al., 2003). Sebbene questi tool si rivelino potenti e la modellazione di uno scenario incidentale attraverso l'ambiente di lavoro risulti molto intuitiva, quando il modello presenta molti input in ingresso essi mostrano due importanti problemi:
1. la risoluzione diventa lunga se non impossibile a causa dell'esplosione nello spazio degli stati;
2. il risultato dei due modelli può spesso non corrispondere per cui risulta arduo comprendere la reale logica implementata con il DFT di cui si vuole conoscere l'esito.

Questo secondo problema (molto annoso per le valutazioni di rischio) è dovuto al fatto che per i DFT non è stato sviluppato fin dal principio un linguaggio semantico ben definito (Coppit et al., 2000), (Boudali et al., 2007) che permetta ai ricercatori lo studio di algoritmi precisi e agli analisti del rischio la comprensione delle logiche di interconnessione delle porte.

In Figura 2.14 viene presentata una sintesi per la classificazione delle caratteristiche essenziali di un albero di guasto, statico e dinamico. Un modello di albero di guasto viene classificato a secondo delle tre seguenti dimensioni:

- distribuzione di probabilità degli eventi base;
- presenza di eventi ripetuti;
- tipo di albero (SFT o DFT).

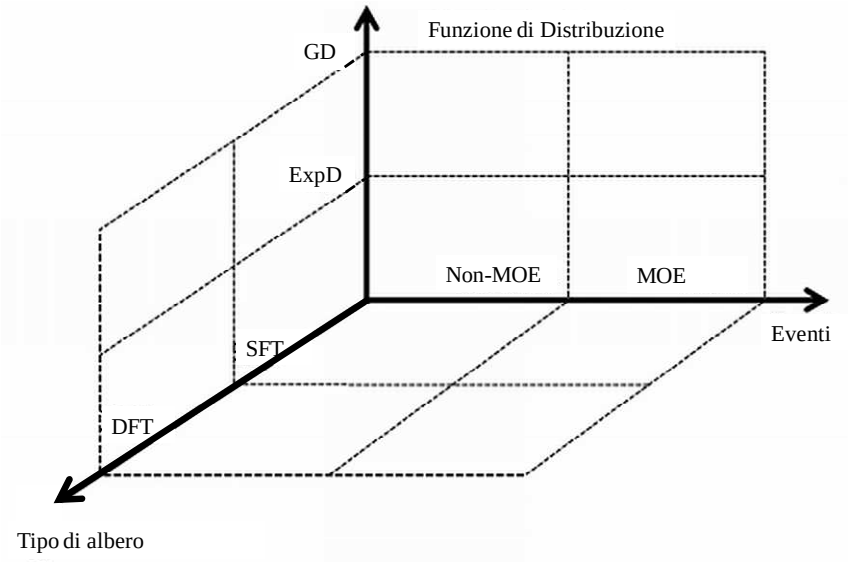


Figura 2.14: Classificazione degli alberi di guasto

Nel panorama letterario disponibile, l'ipotesi fondamentale per la risoluzione dei DFT è che tutti i componenti non siano riparabili (Vesely et al., 1981). Questa ipotesi è estremamente limitativa per i casi reali e per il calcolo di misure importanti come la disponibilità e la probabilità di occorrenza di un evento.

L'approccio classico di risoluzione è basato sulla trasformazione del DFT in un modello di CTMC equivalente. Questa trasformazione, detta "trasformazione diretta", si basa sulla costruzione ricorsiva della matrice dei generatori infinitesimali Q. La matrice Q viene costruita a partire dalla matrice delle transizioni che contiene i

valori delle transizioni di passaggio da uno stato ad un altro (0 se non vi è collegamento fra gli stati). A differenza di una CTMC già modellata, lo svantaggio principale di questo approccio è che la trasformazione diretta di un DFT in un modello nello spazio degli stati può dare luogo al problema dell'esplosione dello spazio degli stati. Infatti, anche piccoli modelli di DFT ad n ingressi danno luogo a matrici Q le cui dimensioni possono essere maggiori di 2^n x 2^n, a causa delle dinamiche tempo dipendenti delle porte. Per esempio, la rappresentazione completa nello spazio degli stati di una PAND a due ingressi con componenti non riparabili è formata da 5 stati come mostrata in Figura 2.15:

1. lo stato iniziale AB è lo stato in cui entrambi i componenti funzionano e il sistema è funzionante;
2. lo stato $\bar{A}B$ è ancora uno stato di funzionamento del sistema, dove il componente A si guasta prima di B. A partire da questo stato è possibile una transizione verso lo stato $\bar{A}\bar{B}$ dovuta al guasto del componente B;
3. lo stato $A\bar{B}$ è ancora uno stato di funzionamento del sistema, dove il componente B si guasta prima di A. A partire da questo stato è possibile una transizione verso lo stato $\bar{A}\bar{B}$ dovuto al guasto del componente A;
4. lo stato $\bar{A}\bar{B}$ può dunque essere raggiunto sia dallo stato $\bar{A}B$ sia dallo stato $A\bar{B}$. Tuttavia l'unico stato di guasto è quello che si raggiunge dallo stato $\bar{A}B$, poiché la condizione di priorità della PAND impone che il guasto del sistema si verifichi solo se A si guasta prima di B.

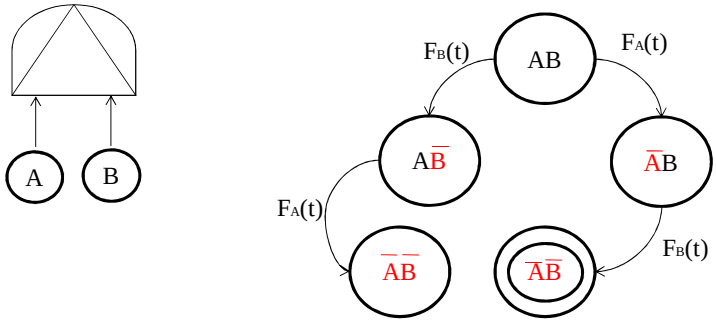


Figura 2.15: Rappresentazione completa nello spazio degli stati di una PAND a due ingressi non riparabili. Lo stato di guasto è quello con il doppio cerchio

Dunque, sistemi complessi formati da molte porte dinamiche possono dare luogo alla creazione di molti stati sollevando problemi sia di carattere tecnico sia di carattere logico. I primi sono legati alla difficoltà di risoluzione di modelli con matrici troppo grandi, difficili da contenere nelle strutture dati delle variabili informatiche dei programmi di risoluzione poiché arrivano ad occupare anche dimensioni dell'ordine dei Gigabyte di memoria.

I problemi di carattere logico dipendono invece dalla consistenza di alcuni stati generati in seguito alla trasformazione diretta. Infatti, da un punto di vista affidabilistico, questi stati possono rivelarsi superflui o illogici. Tale problema è la diretta conseguenza della mancanza di un formalismo semantico universalmente accettato, che potesse permettere agli studiosi di fissare delle regole ben definite. Per esempio, anche la modellazione della PAND di Figura 2.15 può rivelarsi ambigua a causa del duplice carattere dello stato $\bar{A}\bar{B}$; per questo motivo l'utilizzo della tecnica DFT va trattata con molta attenzione poiché possono manifestarsi logiche inaspettate e non previste dal funzionamento reale del sistema studiato.

In appendice, il problema della PAND a due ingressi viene trattato anche per componenti riparabili rivelando come, nell'ambito delle valutazioni degli scenari di rischio, questa porta possa manifestare dei comportamenti non adatti alla modellazione.

In (Merle et al., 2010), invece, viene presentato un elegante approccio basato su un’algebra booleana tempo dipendente per la sintesi della funzione di struttura del DFT; tuttavia essa permette la risoluzione di una classe limitata di DFT, i Priority Dynamic Fault Trees (PDFTs) che contengono qualsiasi composizione di porte statiche e porte PAND con eventi esclusivamente non riparabili che possono essere descritti da qualsiasi distribuzione di probabilità. In questo lavoro, vengono definiti due operatori temporali (*before* e *simultaneous*) grazie ai quali è possibile derivare la formula di struttura del TE per un PDFT, senza passare alla conversione equivalente nello spazio degli stati.

2.9 TECNICA DI GERARCHIZZAZIONE

La gerarchizzazione (o decomposizione) è una tecnica nata per semplificare i tempi e le risorse di calcolo necessarie alla risoluzione di modelli stocastici statici (Chatterjee, 1975), (Modarres, 1979), (Kohda et al., 1989), (Olmos & Wolf, 1996), (Wilson, 1985) che presentano un numero considerevole di nodi.
Questa tecnica comporta la frammentazione del modello statico in frammenti indipendenti che possono essere risolti autonomamente, estrapolando le misure che occorrono per la risoluzione dell'albero originale. L'obiettivo della gerarchizzazione è di semplificare la complessità dell'albero di partenza.
La gerarchizzazione viene realizzata in due fasi:
1. la decomposizione che ha l'obiettivo di individuare le parti indipendenti dell'albero (i sotto modelli) e risolverli;
2. l'aggregazione che ha il compito di mettere insieme i risultati ottenuti nella fase di decomposizione e risolvere il modello aggregato equivalente.

L'applicazione ai modelli dinamici (Gulati & Dugan, 1997), (Anand & Somani, 1998), (Sun & Andrews, 2004), (Ou & Dugan, 2004) si ritrova in quei modelli per cui è possibile trovare moduli dinamici indipendenti.
Questa tecnica ha permesso la risoluzione efficiente di una classe di modelli, ma non ha risolto le problematiche dei modelli complessi che, come vedremo nel successivo capitolo, sono caratterizzati da dinamicità che non possono attraverso la ricerca di moduli dinamici indipendenti.

2.10 APPROCCIO ALLA CLASSIFICAZIONE E ALLA SENSITIVITÀ

Nello studio di un sistema/processo, una delle principali attività della PRA/PSA è la comprensione dei meccanismi che concorrono all'instaurarsi di regimi di lavoro indesiderati al fine di determinare delle strategie di intervento e di contenimento.
La PRA si focalizza in particolare su sistemi, strutture e componenti (SSCs) che contribuiscono al rischio (Cheok et al., 1998), mentre la PSA concentra la propria attenzione sulle SSC che permettono il contenimento dei potenziali rischi. Tali

valutazioni possono essere implementate dalla PRA/PSA attraverso l'uso degli stessi modelli stocastici della dependability già introdotti.

Le misure di importanza (IMs) e l'analisi di sensitività (SA) sono alcune fra le tecniche più usate in questo ambito poiché consentono eseguire indagini più approfondite rispetto alle misure di affidabilità e disponibilità.

Il concetto di Importance Measure fu introdotto da Birnbaum (Birnbaum, 1969) che per primo si rese conto dell'importanza dell'attività di classificazione dei componenti e di valutazione della sensitività nell'ambito affidabilistico.

In letteratura esistono dei contributi che tentano di spiegare come IMs ed SA possano essere usate. Le IMs, sono misure che servono a valutare quali sono i componenti (o i gruppi di componenti) più importanti per l'operatività di un sistema/processo. In (Cheok et al., 1998) la classificazione e la categorizzazione dei SSCs viene effettuata attraverso la:

1. Birnbaum Importance Measure (BIM);
2. risk reduction worth (RRW);
3. risk achievement worth (RAW);
4. misura di Fussell-Vessely (FV),

sottolineando come il problema di una corretta valutazione di importanza dei SSC dipende della corretta impostazione del modello stocastico e dalle ipotesi sugli eventi iniziatori di un TE. Più recentemente in (van der Borst & Schoonakker, 2001) si è evidenziato che l'uso di due sole IMs (la FV e la coppia BIM e RAW) sia sufficiente per gestire ed impostare le attività di manutenzione all'interno di una centrale nucleare.

In (Modarres, 2008) viene discussa la credibilità e l'affidabilità di queste analisi poiché non esistono linee guida standard sull'uso che si può fare di queste misure accessorie. Dal momento che non è possibile fare una generalizzazione di questi risultati, non rimane altro che interpretare, volta per volta, le IMs e la SA in relazione al modello sotto analisi.

Nelle applicazioni pratiche, a seconda del modello stocastico usato gli eventi iniziatori possono essere trattati come:

i. eventi singoli (FT, RBD);

ii. come gruppi di eventi (FT, RBD);
iii. attraverso i parametri che li caratterizzano (modelli nello spazio degli stati).

Il caso (i) è possibile quando non esistono dipendenze fra i componenti del modello (input diversi e possibilmente non ripetuti). Il problema principale di questo approccio è sempre legato al fatto che, in sistemi complessi, le dipendenze fra gli eventi non sono trascurabili.
Per il secondo caso (ii), le IMs di un gruppo di componenti possono essere valutate considerando un intero sottosistema.
Il terzo caso (iii) riguarda il raggruppamento dei parametri tipici di un sistema. Queste valutazioni diventano molto importanti nei sistemi dipendenti poiché l'uso delle IMs, in questi casi, incontra notevoli problematiche. In (Borgonovo & Apostolakis, 2001) viene introdotta una misura differenziale chiamata "Differential Importance Measure" (DIM) che verifica la proprietà di additività rendendo molto più semplice la valutazione delle misure di gruppo.
I modelli combinatoriali offrono tecniche abbastanza consolidate per il calcolo delle IMs (Zang et al., 2002), (Dutuit & Rauzy, 2000), (Veeraraghavan & Trivedi, 1991), ma a causa dei limiti dei modelli stessi, la qualità di indagini di questo genere è opinabile.
Laddove le IMs si focalizzano principalmente sulla struttura fisica del sistema, sia in termini di singoli componenti che di gruppi (Ou & Dugan, 2000), la SA (Sato & Trivedi, 2007), (Gokhale & Trivedi, 2002), (Blake et al., 1988) permette l'analisi dei parametri caratteristici di un sistema, misurando come una variazione di un parametro influenzi le funzionalità del sistema in termini di affidabilità, disponibilità, performance, robustezza. Nel caso dell'inaffidabilità di un sistema F_{sys} possiamo scrivere la sensitività rispetto ad un parametro di un componente (per esempio un tasso di guasto λ) mediante la seguente formulazione:

$$S^{\lambda}_{F_{sys}} = \frac{dF_{sys}}{d\lambda} \qquad (E.2.7)$$

La SA può essere utilizzata sia per i modelli combinatoriali sia per quelli nello spazio degli stati sebbene la valutazione della SA si riveli decisamente più complessa rispetto a quella delle IMs. Infatti, la (*E.2.7*) è utilizzabile solo quando la forma

chiusa della F è nota e in letteratura esistono studi che dimostrano come l'utilizzo di algoritmi numerici possa semplificare (essendo spesso l'unica possibilità) il procedimento di valutazione (Fricks & Trivedi, 2003), (Ou & Dugan, 2000), (Assaf & Dugan, 2004), (Vinod et al., 2003). Infatti, la SA di un sistema rispetto ad un parametro può essere calcolata localmente come la variazione della probabilità di guasto del sistema rispetto alla variazione del parametro. L'approccio tipico fa uso di questa metodica, variando i valori parametro per parametro e sfruttando la potenza dei moderni computer, capaci di effettuare numerosi ricalcoli in poco tempo.
Nello sviluppo sperimentale del capitolo successivo, si applicheranno le definizioni di alcune fra le IMs classiche nell'ambito dei DFT prendendo spunto da (Fricks & Trivedi, 2003) in cui viene mostrata una tecnica che riunisce il calcolo delle IMs sia per modelli combinatoriali che per modelli basati su CTMC, mediante l'uso delle Markov Reward Model (MRM).
La valutazione della SA viene realizzata a partire dalle IMs, sfruttando il teorema di derivazione composta (Ou & Dugan, 2000), limitatamente ad eventi caratterizzati da una distribuzione di probabilità esponenziale negativa.

2.10.1 IMPORTANCE MEASURE

Una rivisitazione esaustiva sulle IMs e sul loro significato può essere rintracciata nei seguenti articoli: (Cheok et al., 1998), (van der Borst & Schoonakker, 2001), (Vinod et al., 2003), (Wang et al., 2004).
La BIM di un componente viene calcolata attraverso la seguente definizione:

$$\boldsymbol{BIM^{X_i}(t) = \frac{\vartheta R_{sys}(t)}{\vartheta r_i(t)} = \frac{\vartheta F_{sys}(t)}{\vartheta f_i(t)}} \qquad \textit{(E.2.8)}$$

La BIM quantifica la variazione con cui l'affidabilità/inaffidabilità di un sistema/processo cambia al variare dell'affidabilità/inaffidabilità di un componente X_i. Alti valori di BIM indicano componenti importanti e un componente X_i è detto critico se la sua BIM_{Xi} è pari ad 1.

Il principale limite della BIM è che il suo valore è indipendente dall'affidabilità del componente per cui più componenti diversi possono esporre la stessa BIM pur avendo un'affidabilità diversa. In un certo senso, la BIM esprime l'importanza di un componente in funzione del suo posizionamento nello schema affidabilistico e per questo motivo non è la misura più adatta per l'attività di classificazione.
Per sistemi coerenti viene provato che la *(E.2.8)* può essere riscritta mediante la seguente riformulazione:

$$BIM^{Xi}(t) = R_{sys}(X_i = 1, X) - R_{sys}(X_i = 0, X) = \\ = F_{sys}(X_i = 0, X) - F_{sys}(X_i = 1, X) \qquad (E.2.9)$$

Come vedremo, dal momento che risulta difficile ricavare la formula di struttura e la forma chiusa per un DFT, l'approccio di calcolo diretto offerto dalla *(E.3.9)* consentirà di ricavare la BIM anche per questi modelli.
Il limite della BIM evidenziato poco sopra viene superato dalla Criticality Importance Factor (CIF) che pesa il valore della BIM mediante il rapporto tra l'inaffidabilità del componente e quella del sistema:

$$CIF^{X_i}(t) = BIM^{X_i}(t)\frac{f_i(t)}{F_{sys}(t)} \qquad (E.2.10)$$

In questo modo, i componenti che hanno la stessa BIM possono essere classificati rispetto a questa misura, dal momento che alti valori di CIF sono associati ai componenti più critici (Borgonovo, 2007).
La misura di Fussel-Vesely (FV) di un componente FV^{Xi} calcola la variazione della probabilità di TE quando un componente è rotto, normalizzata rispetto al valore nominale di guasto:

$$\boldsymbol{FV^{X_i}(t) = \frac{F_{sys}(t) - F_{sys}(X_i=0,X)}{F_{sys}(t)}} \qquad (E.2.11)$$

In (van der Borst & Schoonakker, 2001) è dimostrato che per sistemi coerenti e statici la FV corrisponde esattamente con la CIF.
La Structure Importance Measure (SIM) (Xing & Amari, 2008) misura l'importanza di un componente rispetto al suo posizionamento nell'albero di guasto, senza

considerare la reale affidabilità del componente. Per cui la SIM può essere utilizzata quando non si hanno informazioni sulle caratteristiche dei componenti (Meng, 1996). La SIM del generico componente X_i viene calcolata attraverso la seguente espressione:

$$SIM^{Xi}(t) = \frac{\sum_{\Omega}[F(f_1(t),...,f_{i-1}(t),1_i\ f_{i+1}(t),...,f_n(t)) - F(f_1(t),...,f_{i-1}(t),0_i\ f_{i+1}(t),...,f_n(t))]}{2^{n-1}}$$

dove Ω è l'insieme degli elementi dello spazio campione di tutte le 2^{n-1} combinazioni del possibile vettore dei componenti X. Un metodo alternativo a questa formula si ha utilizzando la seguente espressione:

$$\boldsymbol{SIM^{X_i}(t) = BIM^{X_i}(t|f_k(t) = 0.5), k = 1, 2, \dots, n; k \neq i} \quad \textit{(E.2.12)}$$

assegnando cioè ad ogni componente un valore di probabilità di guasto fissa (pari a 0.5) e calcolando la SIM^{Xi} mediante la BIM^{Xi} (Zang et al., 2002). Inoltre, per modelli statici, la SIM è indipendente dal tempo.

Altre misure di interesse, utilizzate nell'ambito dei sistemi coerenti, sono la RAW (Risk Achievement Worth) e la RRW (Risk Reduction Worth).

L'espressione della RAW del componente X_i è:

$$\boldsymbol{RAW^{X_i}(t) = \frac{F(f_1(t),\dots,f_{i-1}(t),0_i,f_{i+1}(t),\dots,f_n(t))}{F_{sys}(t)}} \quad \textit{(E.2.13)}$$

Questo valore è pari al fattore moltiplicativo con cui l'inaffidabilità del sistema aumenterebbe se il componente, in quell'istante, smettesse di funzionare.

La RRW^{Xi}, complementare alla RAW, è il fattore moltiplicativo con cui l'inaffidabilità del sistema diminuirebbe se il componente i-esimo, all'istante di valutazione, fosse perfettamente funzionante:

$$\boldsymbol{RRW^{X_i}(t) = \frac{F(f_1(t),\dots,f_{i-1}(t),1,f_{i+1}(t),\dots,f_n(t))}{F_{sys}(t)}} \quad \textit{(E.2.14)}$$

In (van der Borst & Schoonakker, 2001) e (Borgonovo, 2007) sono mostrate le relazioni che permettono il passaggio da una misura all'altra; infatti tutte possono essere ricavate a partire dalla BIM.

Esiste anche un legame tra la BIM e la SA che si rivela confrontando la relazione *(E.2.7)* con la *(E.2.8)*: la BIM non è altro che la sensitività dell'affidabilità (o inaffidabilità) di un sistema/processo in funzione dell'affidabilità (o inaffidabilità) di un suo componente.

Mentre le IMs sono più adatte a studiare i cambiamenti di un sistema in funzione dei cambiamenti di una sua parte fisica (di un componente o di un'attività ben precisa), la sensitività viene riferita ai parametri di un sistema.

2.10.2 RELAZIONE TRA LA BIM E LA FTM: IMS PER MODELLI GENERALIZZATI

In letteratura non esistono per i DFT contributi significativi per il calcolo delle IMs.

Come già visto, per un sistema coerente la BIM^{Xi} di un generico componente può essere ottenuta mediante l'equazione *(E.2.9)*, valutando la differenza di affidabilità/inaffidabilità del sistema quando il componente X_i è completamente funzionante e rotto. Con questo metodo, un DFT ad n componenti richiede 2n ricalcoli. Nota la BIM è possibile conoscere le altre IMs, mediante le relazioni matematiche di passaggio.

In (Fricks & Trivedi, 2003) viene sviluppata una tecnica basata sulle MRM per valutare le IMs ed effettuare la SA. Una MRM è una CTMC tale per cui ad ogni stato X_i della catena viene associato un peso detto *reward rate, $r(X_i)$*. In sistemi binari, X_i corrisponde al vettore di n elementi (fatto di 1 e 0) che definiscono lo stato di ogni singolo componente.

Per esempio, nel caso dell'affidabilità/disponibilità, la reward rate $r(X_i)$ da utilizzare è tale che:

$$r(X_i) = 1 \; se \; X_i \text{ è } uno \; stato \; di \; funzionamento$$
$$r(X_i) = 0 \; se \; X_i \text{ è } uno \; stato \; di \; guasto$$

Attraverso le MRM sono possibili interessanti valutazioni che spaziano dal calcolo delle performance di un sistema (associando ad ogni stato un valore di prestazione) a quelle di affidabilità/disponibilità, fino al calcolo delle IMs. Ciò viene realizzato attraverso l'uso di opportune *measure function*, g(·) da applicare alla reward rate di ogni stato. Per esempio, nel caso dell'affidabilità/disponibilità la measure function da usare corrisponde al valore medio atteso della reward: E[r(X)]. In questo caso si può scrivere:

$$\mathrm{R(t)}= \boldsymbol{E[r(X)]} = \sum_{\boldsymbol{X\in\Omega}} \boldsymbol{r(X)\cdot P_X(t)} \qquad \textit{(E.2.15)}$$

dove $P_X(t)$ corrisponde esattamente alla probabilità di transitare per quello stato.
Per un sistema coerente, ad n componenti vale la seguente decomposizione di Shannon:

$$\begin{aligned} r(X) &= X_i \cdot r(1_i, X) + (1 - X_i) \cdot r(0_i, X) \\ &= X_i \cdot [r(1_i, X) - r(0_i, X)] + r(0_i, X) \\ &= \boldsymbol{X_i \cdot \delta_i(X) + \mu_i(X)} \end{aligned} \qquad \textit{(E.2.16)}$$

per ogni $i = 1,..,n$, dove

$$\boldsymbol{\delta_i(X) = \frac{\partial r(X)}{\partial X_i} = r(1_i, X) - r(0_i, X)} \qquad \textit{(E.2.17)}$$

$$\boldsymbol{\mu_i(X) = r(0_i, X)} \qquad \textit{(E.2.18)}$$

Inoltre si ha:

$$\begin{aligned} R(t) = E[r(X)] &= \\ &= [X_i \cdot \delta_i(X) + \mu_i(X)] \\ &= \boldsymbol{R_i(t) \cdot E[\delta_i(X)] + E[\mu_i(X)]} \end{aligned} \qquad \textit{(E.2.19)}$$

Dalla definizione della BIM^{Xi}, usando la *(E.2.8)* e (*E2.19)* si arriva alla seguente:

$$\begin{aligned} \boldsymbol{BIM_i = \frac{\partial R_{sys}}{\partial R_i} = \frac{\partial(E[r(X)])}{\partial R_i} =} & \quad \boldsymbol{E[\delta_i(X)]} \\ & = \boldsymbol{\sum_{X\in\Omega} \delta_i(X) \cdot P_X(t)} \end{aligned} \qquad \textit{(E.2.20)}$$

Attraverso la *(E.2.20)* si ha una comoda formulazione per il calcolo della BIM, come si può vedere nell'esempio di Figura 2.16.

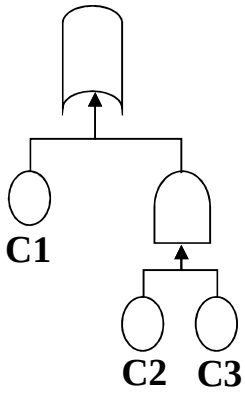


Figura 2.16: Esempio di FT per il calcolo della BIM

Usando l'equazione *(E2.17)* e la Tabella 2.1 è possibile ricavare i δ caratteristici in Tabella 2.2.

Tabella 2.1: Calcolo dei parametri per il metodo di Fricks-Trivedi

x	$(1_1, \mathrm{x})$	$r(1_1, \mathrm{x})$	$(0_1, \mathrm{x})$	$r(0_1, \mathrm{x})$
000	100	0	000	0
001	101	1	001	0
010	110	1	010	0
011	111	1	011	0
100	100	0	000	0
101	101	1	001	0
110	110	1	010	0
111	111	1	011	0

Infine, conoscendo la probabilità di transitare per ognuno degli stati X (vedi Tabella 2.3) è possibile calcolare la BIM mediante l'equazione *(E.2.20)* ottenendo le seguenti equazioni:

$$BIM_i = \begin{cases} R_2(t) + R_3(t) - R_2(t)R_3(t) \\ R_1(t)[1 - R_3(t)] \\ R_1(t)[1 - R_2(t)] \end{cases}$$

Tabella 2.2: Calcolo dei δcaratteristici per il sistema di Figura 2.16

x	$\delta_1(\mathbf{x})$	$\delta_2(\mathbf{x})$	$\delta_3(\mathbf{x})$
000	0	0	0
001	1	0	0
010	1	0	0
011	1	0	0
100	0	1	1
101	1	0	1
110	1	1	0
111	1	0	0
Sum	6	2	2

Il calcolo della BIM attraverso il metodo esposto si rivela interessante nel caso dei modelli dinamici perché non implica la necessità di conoscere la forma chiusa dell'equazione di affidabilità, ma soltanto le probabilità degli stati del sistema.
Tuttavia, un'analisi attenta ci permette di verificare che questa formulazione conduce al valore reale della BIM solo per modelli combinatoriali, diversamente da quanto annunciato in (Fricks & Trivedi, 2003).

Tabella 2.3: probabilità di transitare attraverso gli stati del modello in figura 2.16

x	$r(\mathbf{x})$	$P_{\mathbf{x}}(t)$
000	0	$[1-R_1(t)]\,[1-R_2(t)]\,[1-R_3(t)]$
001	0	$[1-R_1(t)]\,[1-R_2(t)]\,R_3(t)$
010	0	$[1-R_1(t)]\,R_2(t)\,[1-R_3(t)]$
011	0	$[1-R_1(t)]\,R_2(t)R_3(t)$
100	0	$R_1(t)\,[1-R_2(t)]\,[1-R_3(t)]$
101	1	$R_1(t)\,[1-R_2(t)]\,R_3(t)$
110	1	$R_1(t)R_2(t)\,[1-R_3(t)]$
111	1	$R_1(t)R_2(t)R_3(t)$

La controprova di questa affermazione ci viene fornita considerando una porta T-OR a due ingressi, cioè una OR troncata come in Figura 2.17.

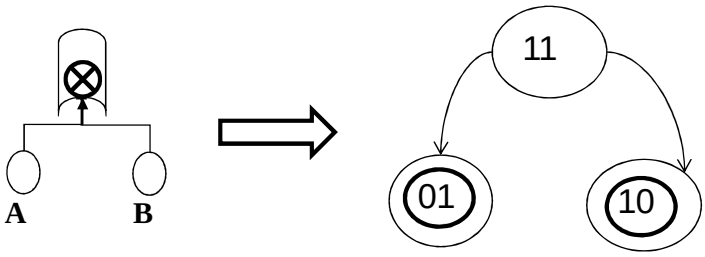


Figura 2.17: OR troncata (T-OR) e rappresentazione nello spazio degli stati

La logica della T-OR è si che quando uno dei due componenti si guasta, il sistema si guasta e non è previsto il guasto di un secondo componente.
Sia per la OR che per la T-OR, l'affidabilità del sistema è data dal prodotto delle singole affidabilità dei componenti:

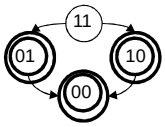
$$R_{sys}(t) = R_A(t) \cdot R_B(t) = P_0(t)$$

La differenza fra le due porte è che per la T-OR la combinazione "guasto-guasto" (che corrisponde allo stato "00", Id=3) non è implementata.
Utilizzando la definizione di BIM *(E.2.8)*, si ha che:

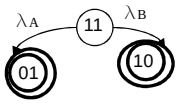
$$BIM_A(t) = \frac{\partial R_{sys}(t)}{\partial R_A(t)} = R_B(t)$$

$$BIM_B(t) = \frac{\partial R_{sys}(t)}{\partial R_B(t)} = R_A(t)$$

Calcolando, invece, la misura di importanza mediante le equazioni *(E.2.20)* e utilizzando la tabella a sinistra (per la porta OR) di Figura 2.18 si ha:



Id	State	r(x)	δ_A	δ_B	$P_x(t)$
0	11	1	1	1	R_AR_B
1	01	0	1	0	F_AR_B
2	10	0	0	1	R_AF_B
3	00	0	0	0	F_AF_B



Id	State	r(x)	δ_A	δ_B	$P_x(t)$
0	11	1	1	1	R_AR_B
1	01	0	1	0	$P_1(t)$
2	10	0	0	1	$P_2(t)$
3	00				0

Figura 2.18: costruzione della tabella di Fricks-Trivedi per la OR e la T-OR

$$BIM_A(t) = P_0 + P_1 = R_A R_B + F_A R_B = R_A R_B + (1 - R_A)R_B = R_B$$

Analogamente

$$BIM_B(t) = P_0 + P_2 = R_A R_B + F_B R_A = R_A R_B + (1 - R_B)R_A = R_A$$

che è lo stesso risultato delle BIM calcolate sopra mediante la definizione.

Tornando alla T-OR, ricordando che l'affidabilità coincide con quella della OR, si può concludere che anche per i componenti della T-OR, le BIM devono coincidere con le precedenti, calcolate per la OR.

Utilizzando invece il metodo FT, mediante la (*E.2.20),* bisogna prima risolvere il modello della CTMC (Figura 2.17) e trovare i valori istante per istante di $P_1(t)$ e $P_2(t)$. Si ha:

$$P_0(t) = e^{-(\lambda_A+\lambda_B)t} = R_A(t)R_B(t)$$

$$P_1(t) = \frac{\lambda_A}{\lambda_A+\lambda_B}\left[1 - e^{-(\lambda_A+\lambda_B)t}\right] = \frac{\lambda_A}{\lambda_A+\lambda_B}[1 - R_A(t)R_B(t)]$$

$$P_2(t) = \frac{\lambda_B}{\lambda_A+\lambda_B}\left[1 - e^{-(\lambda_A+\lambda_B)t}\right] = \frac{\lambda_B}{\lambda_A+\lambda_B}[1 - R_A(t)R_B(t)]$$

Rispetto alla OR, si può notare la differenza dei risultati e concludere che, nei modelli non combinatoriali puri, la misura calcolata con la tecnica di Fricks-Trivedi non corrisponde alla BIM.

Introduciamo quindi una nuova misura, la FTM_i che è quella che si ottiene mediante la *(E.2.20)*.

Infatti, per la T-OR dell'esempio possiamo scrivere:

$$FTM_A(t) = P_0 + P_1 = R_A(t)R_B(t)\left[\frac{\lambda_B}{\lambda_A+\lambda_B}\right] + \frac{\lambda_A}{\lambda_A+\lambda_B}$$

Analogamente,

$$FTM_B(t) = P_0 + P_2 = R_A(t)R_B(t)\left[\frac{\lambda_A}{\lambda_A+\lambda_B}\right] + \frac{\lambda_B}{\lambda_A+\lambda_B}$$

Quindi possiamo concludere che la tecnica in (Fricks & Trivedi, 2003) mediante le MRM conduce al calcolo della BIM solo per modelli combinatoriali puri.

La misura FTM può essere utilizzata per calcolare, mediante le relazioni matematiche di passaggio, le altre IMs (van der Borst & Schoonakker, 2001).

In questo caso avremo:

1. CFT (al posto della CIF);
2. FTRRW (al posto della RRW);
3. FTRAW (al posto della RAW).

2.10.3 MODELLI GERARCHIZZATI

Per modelli gerarchizzati ci si affida alla regola di derivazione composta (Ou & Dugan, 2000), per cui è possibile risalire alla BIM^{Xi} di un componente all'interno di un sottosistema K calcolando prima la BIM^{K} del sottosistema rispetto al modello originale, secondo la sequenza:

$$BIM_{X_i} = \frac{\partial R_{sys}}{\partial R_i} = \frac{\partial R_{sys}}{\partial R_{sub_K}} \cdot \frac{\partial R_{sub_K}}{\partial R_i} \qquad (E.2.21)$$

3. ANALISI DI SISTEMI COMPLESSI

Come già accennato nel precedente capitolo, lo studio di un modello complesso caratterizzato da dipendenze temporali non può essere affrontato senza l'ausilio di tecniche dinamiche. La complessità che si solleva nella risoluzione di un modello del genere non è tuttavia indifferente. Anche la valutazione delle IMs e della SA non è semplice come avviene per i modelli statici tradizionali e il ricorso ad un approccio ad-hoc viene motivato.

Purtroppo, l'uso degli strumenti software automatizzati per la modellazione e il calcolo delle misure della dependability non è sufficiente a risolvere questi modelli.

In questo capitolo, dopo aver introdotto le problematiche legate alla risoluzione di un sistema complesso attraverso l'uso di software dedicati, si estende l'uso della tecnica di gerarchizzazione ai DFT, individuando le ipotesi sotto cui utilizzare l'approccio esatto rispetto a quello approssimato. Al fine di rendere possibile una valutazione più approfondita dei modelli complessi, laddove l'approccio analitico è troppo limitato, si realizza il confronto dei precedenti risultati attraverso delle simulazioni.

Tali confronti consentono di affermare che, allo stato dell'arte, le sole tecniche analitiche non sono in grado di garantire la validità delle valutazioni di rischio legate ai modelli reali. Infatti, da solo, il rigore dell'approccio analitico non è giustificato poiché (sebbene in certi casi consente di giungere a risultati esatti) molto spesso le approssimazioni di risoluzione necessarie in fase di modellazione ne invalidano inevitabilmente la consistenza rispetto alla realtà.

3.1 MODELLAZIONE DI UN SISTEMA COMPLESSO

La modellazione di un sistema per il calcolo di scenari incidentali può essere un'operazione estremamente complessa. Uno strumento di alto livello come quello offerto dalla tecnica dei DFT è di estrema utilità per la fase di definizione dello scenario incidentale.

Il nostro punto di vista è che il DFT è il modello più adeguato poiché, nell'attività di sintesi di uno scenario, permette la descrizione di dinamiche di processo

estremamente complesse, mantenendo una leggibilità molto elevata sia per gli esperti sia per i meno esperti.
Nei casi studio affrontati, il punto di partenza è un modello di DFT che, a seconda della complessità, viene risolto mediante opportune trasformazioni caso per caso.
Durante questa fase del lavoro di tesi si è cercato di sfruttare la combinazione di tecniche di risoluzione dei modelli stocastici, al fine di semplificare la complessità che il DFT puro può presentare.

3.2 SOFTWARE DI CALCOLO

L'analisi dei sistemi complessi richiede anche l'utilizzo di applicazioni software molto potenti. Infatti, dal momento che non è pensabile sviluppare manualmente i calcoli affidabilistici di sistemi complessi, attraverso queste applicazioni l'interfaccia grafica consente un'immediata visualizzazione del modello e la successiva valutazione di rischio risulta semplificata.
Tali software risolvono un modello DFT in due momenti:

1. nella prima fase sviluppano un modello equivalente nello spazio degli stati;
2. nella seconda fase risolvono il modello mediante la teoria di Markov.

Tuttavia, ci si è resi conto che la valutazione di rischio di un sistema complesso non può essere esaustivamente compiuta attraverso l'uso semplicistico dei software in questione.
Infatti, la costruzione del modello di un sistema complesso può non essere immediata perché:

1. la valutazione di alcune misure, come la disponibilità e la probabilità della prima evenienza di un evento, per sistemi dipendenti va oltre le possibilità dei modelli statici;
2. può diventare necessaria la rappresentazione attraverso un modello ibrido (che non è definibile con una sola classe di modelli stocastici);
3. i componenti possono presentare distribuzioni statistiche dei tempi di guasto diverse dalla distribuzione di Poisson.

La qualità della soluzione finale dipende dall'accuratezza della trasformazione e dalla conseguente risoluzione.

I software di calcolo possono dunque essere classificati in funzione di queste caratteristiche e degli algoritmi usati per assolvere a queste funzioni.

In questo lavoro di tesi sono stati utilizzati tre fra i più importanti programmi per l'analisi del rischio attualmente in circolazione:

1. Relex ®, prodotto dalla PTC;
2. Galileo ®, sviluppato dalla University of Virginia;
3. SHARPE ®, proposto dalla Duke University.

L'analisi di tali pacchetti applicativi ha consentito di conoscere lo stato dell'arte degli strumenti software oggi disponibili per le analisi affidabilistiche, mettendo in evidenza sia le potenzialità di ciascuno strumento si i relaviti limiti.

Nella Tabella 3.1 sono sintetizzati il tipo di modellazione dinamica che ogni tool consente di realizzare specificando se per la risoluzione esso adotta strumenti automatici o manuali di gerarchizzazione e indicando le misure che il tool è in grado di fornire.

Tabella 3.1: Caratteristiche dei pacchetti software per l'analisi di affidabilità

TOOL	MODELLAZIONE DINAMICA	GERARCH	DISTR. PROB.			MISURE CALCOLATE				
			EXP	WEIB	LOG	R	A	IM	SA	OM
RELEX	DFT WIZARD HCTMC	Manuale	S	N	N	S	S	S	N	S
GALILEO	DFT WIZARD	Automatica	S	S	S	S	N	S	S	N
SHARPE	HCTMC GSMP MRGP SPN	Manuale	S	S	S	S	S	S	S	S

Altre informazioni sono fornite qui di seguito:

1. Relex®: fornisce un ambiente di lavoro altamente intuitivo per la modellizzazione dei DFT. Il motore di Relex® permette il calcolo delle misure di affidabilità e disponibilità e può risolvere alberi che presentano MOE; il limite di Relex® è che gli input di un DFT possono essere caratterizzati solo attraverso la distribuzione esponenziale negativa e la gerarchizzazione deve essere realizzata manualmente assemblando modelli ibridi che fanno uso dei modelli stocastici quali gli RBD e le HCTMC;
2. Galileo: anche questo tool, in modo analogo al Relex®, fornisce un ambiente di lavoro intuitivo per la costruzione del DFT, basato su Microsoft Visio®. Galileo permette il calcolo della sola misura di affidabilità; i MOEs non sono permessi ma, diversamente da Relex®, gli input del DFT possono essere caratterizzati anche dalle distribuzioni di Weibull e Log-Normali. La gerarchizzazione è gestita automaticamente in quanto i moduli dinamici sono convertiti automaticamente in HCTMC o in reti di Bayes;
3. SHARPE: in questo tool sono offerti molteplici strumenti per le valutazioni di rischio e delle prestazioni di un sistema. Sono supportati modelli di tipo SFT, i Reliability Graph, gli RBD e modelli nello spazio degli stati come le HCTMC, le GSMP e le MRGP. La modellazione automatizzata dei DFT non è implementata per cui tali modelli devono essere risolti costruendo manualmente un modello equivalente nello spazio degli stati. Il software permette quindi lo sviluppo di modelli ibridi grazie a delle funzioni per i modelli gerarchizzati che consentono il calcolo delle misure dell'affidabilità, disponibilità e delle IMs.

3.3 RISOLUZIONE DI UN MODELLO COMPLESSO

Per la risoluzione del DFT l'analisi di fattibilità del modello è molto importante. In questa fase bisogna essere in grado di capire se la risoluzione può essere effettuata attraverso i software elencati. Infatti, se le caratteristiche del DFT (struttura, ingressi e misure da valutare), vedi Figura 2.14, non si confanno alle specifiche dei tool, la risoluzione diretta attraverso il software è improponibile.

Per questo motivo, la risoluzione di un DFT può essere eseguita attraverso diversi approcci che vengono implementati caso per caso: non esiste un approccio standard ed è spesso necessario attuare delle trasformazioni del modello, se non addirittura delle semplificazioni dell'albero originale (Figura 3.1). Dal momento che ogni operazione di questo tipo si può ripercuotere sulla precisione finale del risultato, anche l'analisi del processo di risoluzione e della degradazione del modello originale fa parte della valutazione finale dei risultati ottenuti.

Per i casi sperimentali trattati, la tecnica di risoluzione più utilizzata è basata sulla conversione diretta in una CTMC e, laddove i casi sono favorevoli, viene sfruttato il paradigma della gerarchizzazione. Infatti, una riduzione della grandezza della matrice dei generatori infinitesimali Q si può ottenere se il modello di DFT viene reso più snello nella fase precedente alla trasformazione in HCTMC.

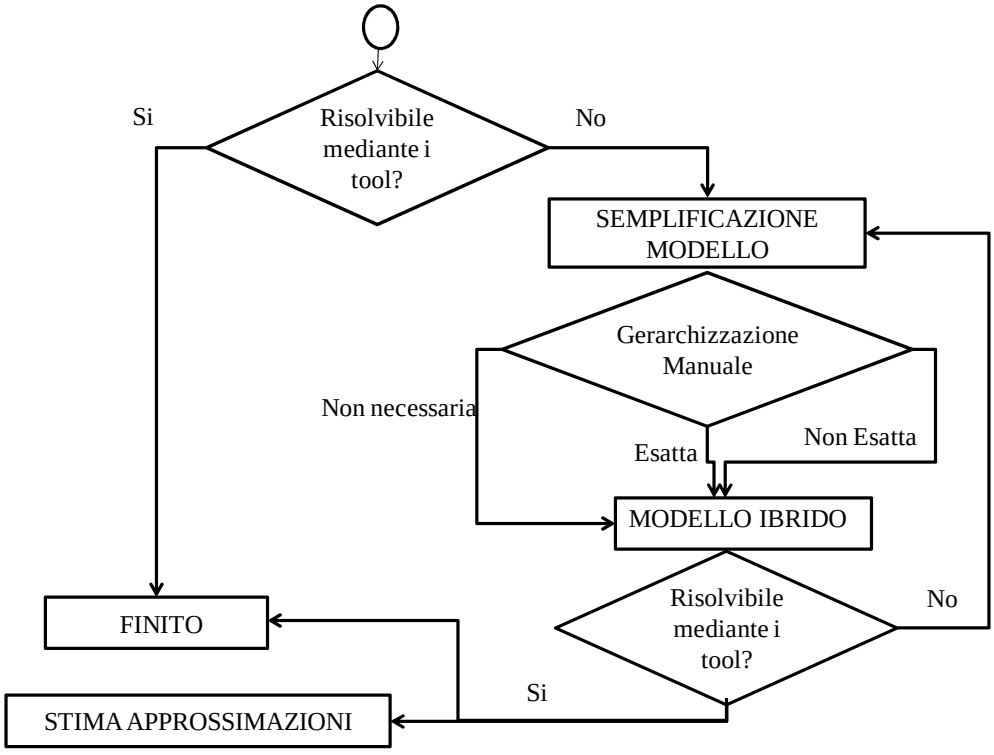


Figura 3.1: Schema di flusso per la risoluzione di un DFT

3.4 GERARCHIZZAZIONE DI UN DFT

La gerarchizzazione di un modello di DFT può essere realizzata in modo del tutto analogo a quanto visto per un SFT (Figura 3.2): attraverso un'analisi top-down si individuano le sezioni dell'albero indipendenti e una volta risolti tali sottoblocchi si passa alla riaggregazione in un albero equivalente, il DFT*, la cui rappresentazione nello spazio degli stati risulta notevolmente semplificata. Per chiarezza chiameremo "sottomodelli composti" le parti dell'albero originale che sono state sottoposte ad aggregazione e "modello gerarchizzato" la rappresentazione finale dell'albero equivalente (il DFT*) alla fine del processo di gerarchizzazione.

Il problema principale della gerarchizzazione nei DFT è che la fase di decomposizione non è immediata come quella che si realizza per gli SFT, poiché le dipendenze temporali fra i componenti del DFT introducono delle complicazioni non indifferenti.

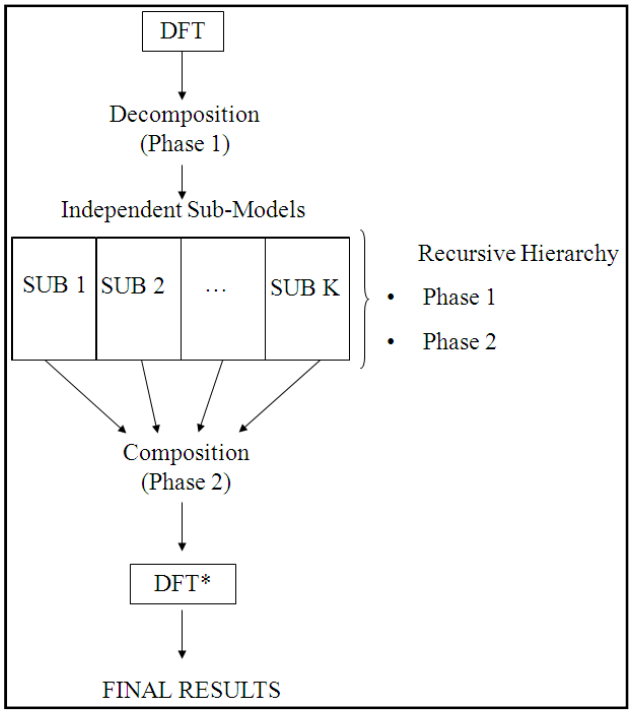


Figura 3.2: **la procedura di gerarchizzazione ricorsiva secondo le due fasi di decomposizione e riaggregazione del modello**

Questo aspetto è preponderante per i DFT che presentano una sequenza in cascata di porte dinamiche. Per questo motivo, l'uso corretto della gerarchizzazione deve essere

valutato con attenzione al fine di ottenere dei risultati congrui con il modello originale che si analizza.

Per un SFT l'approccio risulta valido anche per il calcolo della disponibilità di un sistema poiché la dipendenza temporale non è sentita ai livelli alti dell'albero. Per un DFT la gerarchizzazione è completamente inappropriata per il calcolo della disponibilità e dell'occorrenza di un evento indesiderato poiché la riparazione di un componente ai livelli più bassi si ripercuote sulle dinamiche sviluppate ai livelli più alti. Infatti, un evento riparabile fa sì che il DFT non sia più un DAG. In ogni caso, data la complessità del modello risultante, lo studio della tecnica di gerarchizzazione per un DFT a componenti riparabili non è stato approfondito.

In un generico FT le dipendenze fra gli eventi sono dovute alla struttura dell'albero e all'evoluzione dello scenario (lo stato del sistema) che si sviluppa attraversando l'albero dal basso verso l'alto, fino alla porta di TE. Ma, dato che una porta dinamica considera anche la dimensione temporale, il comportamento della porta dipende anche dalle dipendenze temporali che si sviluppano in precedenza (cioè quelle dipendenze modellate dalle porte di livello più basso, siano esse statiche o dinamiche). Dal punto di vista della gerarchizzazione, quindi, non è possibile individuare moduli indipendenti al di sotto di una porta dinamica.

Sulla base di lavori precedenti (Gulati & Dugan, 1997), (Anand & Somani, 1998), è possibile distinguere due tipi di approccio diverso alla gerarchizzazione:

1. l'**approccio esatto** *(strong approach)* che considera qualsiasi dipendenza temporale ad ogni fase del processo di gerarchizzazione e restituisce un risultato esatto;
2. l'**approccio approssimato** *(weak approach)* che, durante il processo di gerarchizzazione, trascura scomode dipendenze temporali pervenendo ad un risultato non esatto.

In Figura 3.3 è mostrato un modello molto utilizzato di schema di DFT per il calcolo dell'affidabilità e della disponibilità (Gulati & Dugan, 1997), (Sun & Andrews, 2004), (Ou & Dugan, 2004). Questa tipologia di DFT non presenta MOE; inoltre le porte dinamiche sono tutte poste ai livelli più bassi. I blocchi in bianco possono essere pensati come sotto modelli di livello intermedio dell'albero che contengono solo porte statiche; i blocchi in grigio sono invece i sotto modelli del livello più basso

per i quali non è prevista nessuna ulteriore decomposizione. Questi ultimi rappresentano dunque degli SFT completamente ridotti oppure dei DFT che andranno risolti mediante un equivalente modello nello spazio degli stati. Per questo tipo di DFT, è possibile utilizzare l'approccio gerarchico esatto. Infatti, le dipendenze temporali vengono risolte tutte ai livelli bassi dell'albero e la loro dinamicità non si propaga verso le porte statiche dei livelli più alti. Nel modello gerarchizzato dell'albero (il DFT* equivalente) i sottomodelli vengono rimpiazzati con degli eventi equivalenti che, ad ogni istante, esprimono la stessa probabilità di guasto del sottomodello di partenza.

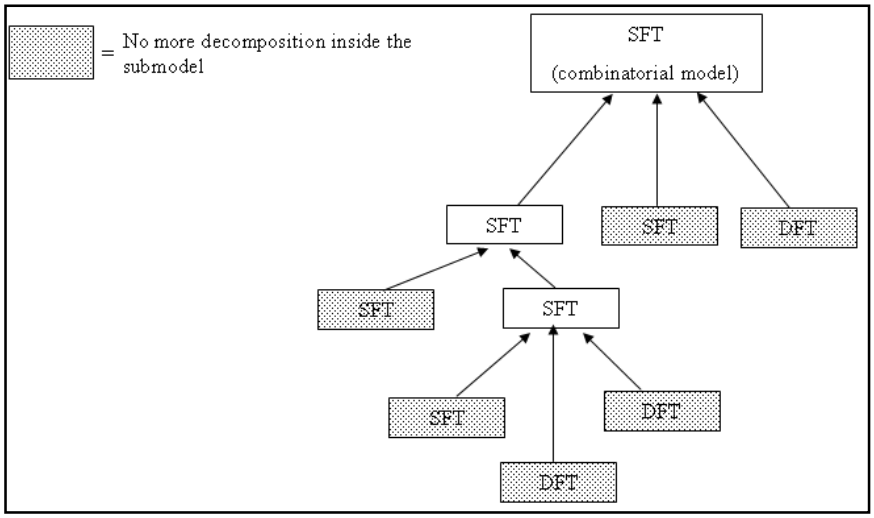


Figura 3.3: classe di DFT gerarchizzabile mediante approccio esatto

Con questa trasformazione, il DFT gerarchizzato viene trasformato in un SFT che può essere risolto attraverso la logica booleana delle tecniche statiche, fornendo dei risultati esatti.

Per questo tipo di DFT, la risoluzione attraverso i software è abbastanza immediata: Relex® e Galileo possono processare il DFT* (quest'ultimo anche per ingressi con una distribuzione di probabilità log-normale o di Weibull), mentre per SHARPE diventa necessario sviluppare la CTMC dei sottomodelli del DFT* equivalente e incapsulare successivamente i risultati nel modello gerarchizzato.

Quando le porte dinamiche sono disposte in cascata, questo approccio diventa impraticabile a meno di isolare un sottomodello a partire dalla porta dinamica di più

alto livello. In (Merle et al., 2010) viene proposta un'elegante tecnica basata sua una formula di struttura tempo dipendente valida per il calcolo dell'affidabilità di un DFT che presenta MOE, caratterizzato solamente da porte dinamiche PAND. Allo stato dell'arte, l'approccio più efficace è la trasformazione del DFT in un modello nello spazio degli stati.

La Figura 3.4 mostra due DFT simili, il modello DFT* gerarchizzato e le relative rappresentazioni nello spazio degli stati. Appare chiara la semplificazione che si ottiene dopo l'operazione di gerarchizzazione. Il sottomodello composto del caso (a) (il sistema sotto la porta OR) differisce dal sottomodello composto del caso (b) (sotto la porta AND) e tale diversità viene mantenuta nel modello gerarchizzato del DFT*; infatti, il componente C equivalente del DFT* viene caratterizzato dai parametri estratti a seguito dell'operazione di gerarchizzazione.

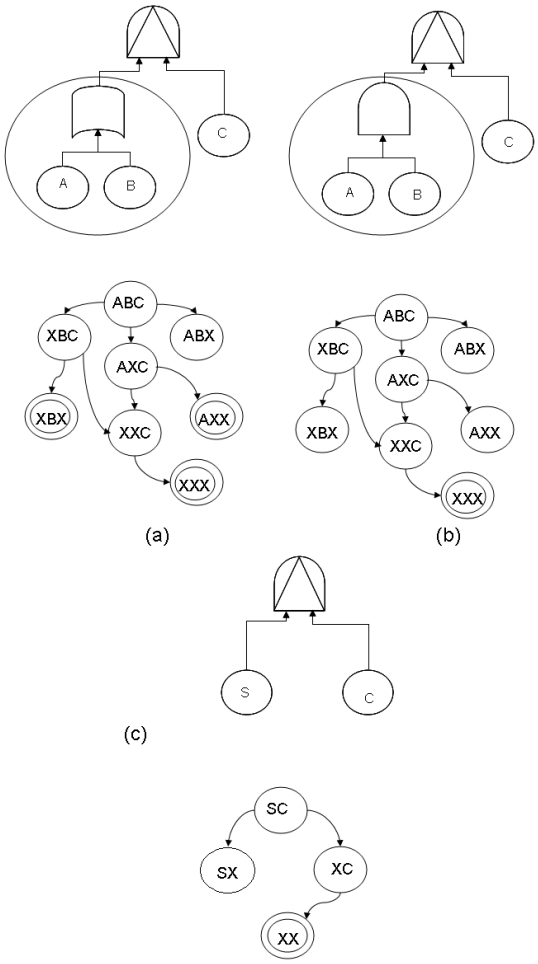


Figura 3.4: due differenti DFTs (a, b) che adottano lo stesso modello gerarchizzato (c); (a) verifica una gerarchizzazione esatta, (b) quella inesatta.

In termini di CDF, il componente C viene descritto da una distribuzione di probabilità generalizzata che caratterizza istante per istante il sottomodello gerarchizzato: queste funzioni tengono traccia delle dipendenze temporali che caratterizzano il sottosistema, in modo da soddisfare l'approccio esatto alla gerarchizzazione. Lo svantaggio di questa metodologia sta nel fatto che il modello gerarchizzato non è più una CTMC: in questi casi l'uso dei software non è immediato a meno di opportuni aggiustamenti nel modello. Infatti:

- Relex® permette solo l'uso di input con CDF di tipo esponenziale negativo;
- Galileo non consente l'uso di distribuzioni generalizzate;
- SHARPE può rappresentare una NHCTMC, ma richiede la conversione manuale del modello gerarchizzato nello spazio degli stati. Inoltre, SHARPE può risolvere soltanto una certa classe di NHCTMC, per cui una generalizzazione non è immediata. Infatti, la distribuzione generalizzata (del modello equivalente) può essere usata solo nella transizione dallo stato iniziale verso uno stato successivo.

L'esempio in Figura 3.5 può aiutare a comprendere quanto espresso: una PAND a 2 ingressi è modellata secondo le due possibili permutazioni degli ingressi. Nel primo caso (a), si assuma che uno degli eventi input, il BE_G, sia l'evento di un sottomodello composto G (già sottoposto a gerarchizzazione) caratterizzato da una distribuzione generalizzata di probabilità, G(t); l'altro evento base, il BE_E, sia invece caratterizzato da una CDF di tipo esponenziale negativa di parametro λ.
Utilizzando la gerarchizzazione esatta, il modello più appropriato è in questo caso una GSMP in cui la G(t) estrapolata come CDF del sottomodello composto G viene usata per descrivere la transizione di stato della GSMP che modella il sistema gerarchizzato in Figura 3.5(a); per questo tipo di modello SHARPE può effettuare, attraverso il "competing process" (Sahner et al., 1996), la valutazione del modello. Infatti, la G(t) estratta viene usata a partire dal tempo t=0, conoscendo la condizione iniziale di affidabilità del sottosistema (R(t) = 1).

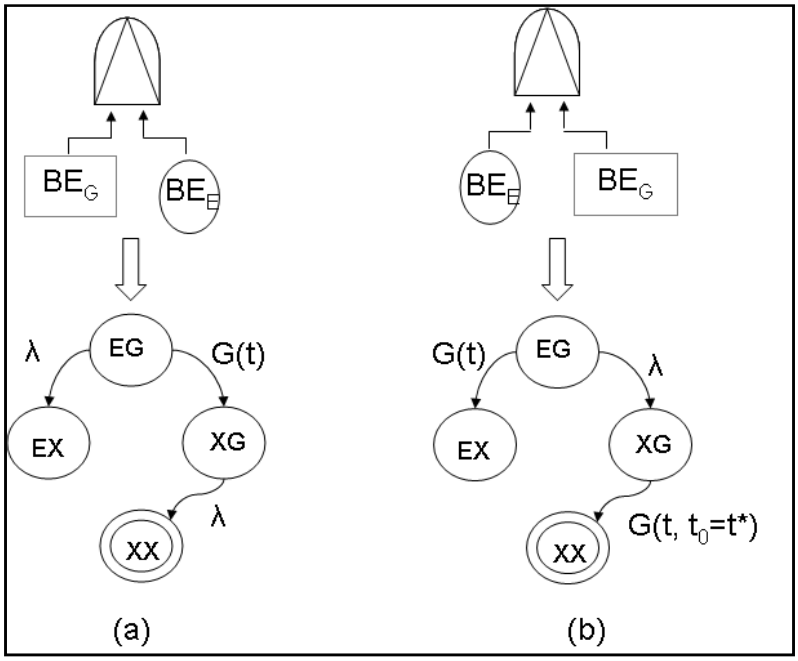


Figura 3.5: (a) GSMP per una PAND a 2 ingressi; (b) GSMP della PAND a 2 ingressi di (a) con gli ingressi permutati di ordine

Nella seconda modellizzazione Figura 3.5(b), dove il modello composto G è usato come secondo input alla PAND, la funzione G(t, t_0=t*) che modella la transizione dallo stato "XG" allo stato "XX" è sconosciuta poiché l'istante t* che trascorre dopo la transizione dallo stato "EG" allo stato "XG" è aleatorio.

L'approccio inesatto alla gerarchia può risolvere questo problema, poiché approssima il comportamento statistico di tutto il sottosistema composto mediante una distribuzione di probabilità esponenziale negativa, caratterizzata da un tasso di guasto equivalente (Anand & Somani, 1998).

Attraverso gli esempi di Figura 3.4, si mostra la differenza dei risultati analitici tra l'approccio esatto e quello inesatto.

Si assuma che il tempo di guasto di ogni BE sia esponenzialmente distribuito. Attraverso queste ipotesi, si ha:

$$\boldsymbol{R_A(t) = e^{-\lambda_A t};\ F_A(t) = 1 - e^{-\lambda_A t}} \qquad \textit{(E.3.1)}$$

$$\boldsymbol{R_B(t) = e^{-\lambda_B t};\ F_B(t) = 1 - e^{-\lambda_B t}} \qquad \textit{(E.3.2)}$$

Modello (Figura 3.4a):

In questo caso il sottosistema composto è incapsulato al di sotto di una porta OR. Per cui:

$$\boldsymbol{R_{OR}(t) = R_A(t) \cdot R_B(t) = e^{-(\lambda_A+\lambda_B)t}} \qquad (E.3.3)$$

Dunque, la CDF del sottosistema composto è ancora caratterizzata da una distribuzione esponenziale negativa pura di parametro $\lambda_{eq} = \lambda_S = \lambda_A + \lambda_B$. Il modello di DFT* gerarchizzato (Figura 3.4c) viene caratterizzato da questo parametro λ_{eq} e i risultati finali corrisponderanno esattamente a quelli del modello DFT originale.

La soluzione in forma chiusa per il modello (c) viene facilmente calcolato attraverso le equazioni di Chapman-Kolmogorov (Trivedi, 2002):

$$\boldsymbol{F_{sys}(t) = P_{XX}(t) = \frac{\lambda_S}{\lambda_S+\lambda_C} - e^{-\lambda_C t}(1 - \frac{\lambda_C}{\lambda_S+\lambda_C} e^{-\lambda_S t})} \qquad (E.3.4)$$

Modello (Figura 3.4b):

La tecnica di gerarchizzazione viene applicata sul sottomodello incapsulato sotto la porta AND. In questo caso si ha:

$$\begin{aligned} F_{AND}(t) &= F_A(t) \cdot F_S(t) = \\ &= (1 - R_A(t)) \cdot (1 - R_S(t)) = \\ &= 1 - R_S(t) - R_A(t) + R_A(t)R_S(t) \end{aligned} \qquad (E.3.5)$$

Diversamente dal modello (Figura 3.4a), la CDF del sottosistema è esponomiale: sebbene l'equivalenza del parametro caratteristico utilizzata in (Figura 3.4a) non è applicabile, in (Anand & Somani, 1998) viene proposta l'approssimazione attraverso una CDF esponenziale negativa; in questo caso, l'approccio di gerarchizzazione dà luogo ad un risultato inesatto. Per trovare il tasso di guasto equivalente, $h(t) = \lambda_{eq}$, da usare nel modello gerarchizzato DFT* (Figura 3.4c) si possono seguire due approcci:

1. attraverso l'inversione della CDF esponenziale negativa, si calcola il tasso di guasto istantaneo all'istante T*:

$$h(T^*) = \lambda_{eq} = -\frac{\ln R(T^*)}{T^*} \qquad (E.3.6)$$

dove T* è il tempo a cui si vuole calcolare l'affidabilità del sistema DFT originale. Diversamente da un modello di DFT puramente statico, il calcolo del tasso di guasto equivalente λ_S può dar luogo a ulteriori approssimazioni. Una volta noto il λ_S, l'affidabilità del sistema gerarchizzato (Figura 3.4c) viene calcolata ponendo $\lambda_{eq} = \lambda_S$ e usando l'equazione *(E.3.4)* al tempo T*;

2. il secondo approccio prevede l'uso dell'inverso del MTTF come tasso di guasto equivalente. Questa soluzione sfrutta la proprietà della distribuzione esponenziale negativa per cui l'inverso del MTTF corrisponde esattamente al tasso di guasto. Partendo dall'equazione per il calcolo del MTTF:

$$MTTF = \int_0^\infty R(t)\,dt$$

si ha,

$$\boldsymbol{MTTF_{AND} = \int_0^\infty R_{AND}(t)\,dt = \frac{\lambda_A^2+\lambda_B^2+\lambda_A\,\lambda_B}{\lambda_A^2\lambda_B+\lambda_B^2\lambda_A}} \qquad \textit{(E.3.7)}$$

da cui:

$$\boldsymbol{\lambda_{eq} = \frac{1}{MTTF_{AND}} = \frac{\lambda_A^2\lambda_B+\lambda_B^2\lambda_A}{\lambda_A^2+\lambda_B^2+\lambda_A\,\lambda_B}} \qquad \textit{(E.3.8)}$$

In Tabella 3.2 sono mostrati gli errori relativi con l'uso dei due approcci di gerarchia inesatta W1 e W2.

Tabella 3.2: Inaffidabilità DFT di Figura 3.4b (analitico) ed errori introdotti con l'approccio gerarchico non esatto W1 e W2 per il DFT equivalente (Figura3.4c)

λT_m	Unreliability				
	Analytic	W1	Δe_{rel}	W 2	Δe_{rel}
10	$3.33\text{x}10^{-1}$	$5.51\text{x}10^{-1}$	83%	$3.99\text{x}10^{-1}$	20%
1	$8.39\text{x}10^{-2}$	$1.98\text{x}10^{-1}$	136%	$1.45\text{x}10^{-2}$	73%
10^{-1}	$2.87\text{x}10^{-4}$	$4.52\text{x}10^{-3}$	1478%	$3.05\text{x}10^{-3}$	962%
10^{-2}	$3.28\text{x}10^{-7}$	$4.95\text{x}10^{-5}$	14960%	$3.31\text{x}10^{-5}$	9973%
10^{-3}	$3.31\text{x}10^{-10}$	$4.94\text{x}10^{-9}$	1394%	$3.33\text{x}10^{-7}$	100515%

Si è supposto che tutti i BE avessero lo stesso tasso di guasto λ e, come si può vedere dalla Tabella 3.2, il parametro caratteristico della distribuzione esponenziale negativa viene considerato moltiplicato con il tempo di missione t.
Il calcolo del λ_s del modello gerarchizzato (Figura 3.4c) viene effettuato per entrambi gli approcci W1 e W2, rispettivamente mediante le equazioni *(E.3.6)* e (*E.3.8)*.
Per entrambi gli approcci approssimati, è possibile notare come generalmente all'aumentare del parametro λt corrisponde un aumento dell'errore percentuale che risulta non trascurabile per applicazioni reali. Infatti, in questi casi il parametro λt deve mantenersi basso affinché il sistema reale esponga un'affidabilità elevata.
Sebbene l'approccio non esatto fornisca una risoluzione molto sommaria consentendo l'uso di modelli poi implementabili attraverso i programmi software, il suo impiego va, dunque, valutato con molta attenzione.

3.4.1 CONSIDERAZIONI SULL'USO DELLA GERARCHIZZAZIONE

Come visto, la gerarchizzazione consente di rendere più snello - e quindi processabile da un software dedicato - un modello molto complesso.
Si sono individuati due approcci: quello esatto e quello approssimato. Purtroppo la risoluzione di un modello attraverso l'approccio analitico esatto non è sempre fattibile nei DFT, mediante i tool introdotti. I limiti di questi tool sono essenzialmente legati alla tecnica di risoluzione nello spazio degli stati mediante delle CTMC, per le quali non è possibile utilizzare delle distribuzioni generalizzate.
Le ipotesi sotto cui la gerarchizzazione di un DFT è esatta sono:

H1. i tempi di guasto degli input delle porte del DFT possono essere descritti soltanto mediante una CDF esponenziale negativa;
H2. gli input delle porte non sono riparabili;
H3. il sottomodello deve essere descritto dalla sua reale distribuzione di probabilità;

In pratica, queste ipotesi comportano le ulteriori seguenti riflessioni:

a. sottomodelli composti che afferiscono ad una porta dinamica possono essere descritti soltanto da una CDF esponenziale negativa. Si ha che l'unica possibilità in cui la CDF di sottosistema risulta esatta è che il sottosistema in questione sia formato soltanto da porte OR;

b. sottomodelli composti che afferiscono a porte statiche possono essere descritti mediante la probabilità di guasto istante per istante. In tali casi, si ha che la gerarchizzazione viene realizzata secondo l'approccio classico, considerando un evento equivalente nell'albero gerarchizzato DFT*; il sottosistema composto viene sostituito con un evento che espone una probabilità costante di guasto calcolata al generico istante t.

Sempre sotto le ipotesi H1, H2 e H3 l'utilizzo della gerarchizzazione approssimata si realizza quando la trasformazione di un sistema composto in una CTMC è troppo esosa e quando delle porte dinamiche si trovano a dei livelli alti del DFT. In questo caso si può associare ad un sottosistema composto (al di sotto della porta dinamica) una funzione di probabilità di sistema equivalente di tipo esponenziale negativa (Anand & Somani, 1998), il cui parametro viene stabilito risolvendo il sottosistema in questione. Questa tecnica, per costruzione, può rivelarsi troppo approssimativa poiché non è sempre conveniente associare a una distribuzione generalizzata qualunque (che è in realtà una composizione, non solo lineare, di esponenziali negative) il comportamento di una esponenziale negativa.
Le phase-type distribution (Neuts, 1983) cercano di ovviare a questo problema esplodendo un sottosistema (che espone una distribuzione generalizzata) in una CTMC la cui complessità varia al variare del grado di precisione da ottenere. In questi casi passaggi di transizione sono tutti mediati attraverso dei tassi di transizione determinati attraverso l'algoritmo delle phase-type. Sebbene interessante, il ricorso a questa tecnica può comportare i classici problemi di esplosione nello spazio degli stati; inoltre, dal momento che si fa sempre ricorso ad una HCTMC equivalente, questo approccio risulta comunque non funzionale alla modellazione mediante i tool di affidabilità presi a riferimento.

3.5 SIMULAZIONE AD EVENTI DISCRETI

Quando la valutazione analitica di un FT diventa ostica e la gerarchizzazione non introduce delle significative migliorie per la semplificazione del modello, può essere utile condurre una campagna di simulazioni (Durga & al, 2007).

L'approccio simulativo nasce con l'intento di rendere possibili le valutazioni in ambito affidabilistico per sistemi sofisticati. L'idea alla base di questo approccio è la simulazione dei reali scenari di processo; la valutazione dell'affidabilità viene effettuata attraverso la definizione classica di probabilità, basata sul rapporto tra eventi favorevoli sul numero totale di eventi.

Grazie ai moderni sistemi di calcolo capaci di effettuare un elevato numero di simulazioni, le valutazioni delle misure affidabilistiche per sistemi complessi possono essere condotte mediante un approccio basato sulla tecnica Monte Carlo (Goldfeld & Dubi, 1987) ottenendo misure molto prossime a quelle analitiche. Il metodo Monte Carlo viene implementato realizzando un numero considerevole di simulazioni del sistema e misurando il valore della grandezza di interesse; tali misure corrisponderanno all'insieme dei punti di un "random walk" che appartengono solo alle configurazioni ammissibili nello spazio delle fasi del sistema.

Alcuni dei tool simulativi più conosciuti per le analisi di affidabilità sono il BlockSim® e OpenFT-A®.

Rispetto agli approcci simulativi tradizionali, in questo lavoro di tesi, non viene simulata l'evoluzione del sistema attraverso lo spazio di stato (random walk), ma si considera ogni input del FT come un'entità base. Per ognuno di questi input viene calcolato il tempo di guasto e queste informazioni sono passate alle porte dei livelli più alti. Lo stato della porta viene determinato attraverso delle operazioni logiche (tempo-dipendenti) fino a risalire completamente lungo l'albero. Per la realizzazione dei modelli simulativi si è fatto uso di Excel® che offre delle funzioni primitive molto intuitive, utili alla verifica delle logiche tipiche delle porte dei FT. Il tipo di simulazione che si realizza prende il nome di simulazione ad eventi discreti. Questo approccio fa uso dei risultati dell'estrazione mediante simulazione Monte Carlo classica. Infatti, tali risultati alimentano un modello (in questo caso il FT) che ha delle logiche dinamiche proprie. Sono queste logiche che fanno evolvere il sistema.

Per creare un modello di simulazione ad eventi discreti in Excel® vengono seguiti i seguenti passi:

a) si identifica l'orizzonte temporale T_m;

b) per ogni input del modello si identifica il comportamento statistico (natura dei guasti descritta da una propria CDF) e si codifica il loro comportamento campionato nel tempo, indipendentemente dal comportamento degli altri;

c) si implementano le relazioni tra gli input e le porte, attraverso le regole che caratterizzano le porte dinamiche e statiche;

d) si realizzano dei buffer per contenere i risultati (ad ogni iterazione) delle dinamiche delle varie porte (fino alla porte di TE) e si verifica la convergenza dei vari risultati attraverso un test di errore.

Per la scelta del tempo di campionamento, una volta determinato il grado di precisione con cui realizzare le simulazioni (cioè il numero di campioni N), viene suddiviso l'intervallo [0, T_m] mediante la formula: T_m /N.

Il tempo di guasto di un ingresso viene calcolato per ogni simulazione mediante la relazione inversa della CDF (Tabella 3.3), realizzando il motore della simulazione Monte Carlo. Infatti, attraverso la funzione di generazione dei numeri casuali (la RAND), si estrae un numero pseudo-casuale uniformemente distribuito tra [0,1] usato per ricavare, mediante la formula inversa, il tempo di guasto del componente.

Tabella 3.3: funzioni Excel® utilizzate per la codifica del motore Monte Carlo per le simulazioni

GD	F(TBEi)	$TBEi=F(TBEi)^{-1}$
Exponential	+ RAND()	- LN(1-RAND())/TBEim
Gaussian	+ RAND()	+ NORMINV(RAND();$TBEi_m$;σ_{TBEi})
Costant	+ RAND()	+ RAND()*$TBEi_{max}$

La Figura 3.6 mostra l'approccio ad eventi discreti: un numero casuale è estratto e usato come valore della CDF (il codominio). Proiettando il valore del codominio lungo l'asse dei tempi si ottiene il tempo di guasto dell'evento.

In Figura 3.6 è mostrata l'implementazione della regola di una PAND a due ingressi, attraverso due CDF differenti: la normale e l'esponenziale negativa. Uno dei vantaggi dell'approccio ad eventi discreti è legato al fatto che, una volta che il modello logico

e dinamico dell'albero è implementato, le CDF associate ad ogni evento possono essere modificate senza dover più alterare la struttura dell'albero.

Tuttavia, come detto, questo sistema richiede l'implementazione ad-hoc per ogni scenario di guasto, in cui ogni porta va codificata correttamente secondo la propria logica di funzionamento.

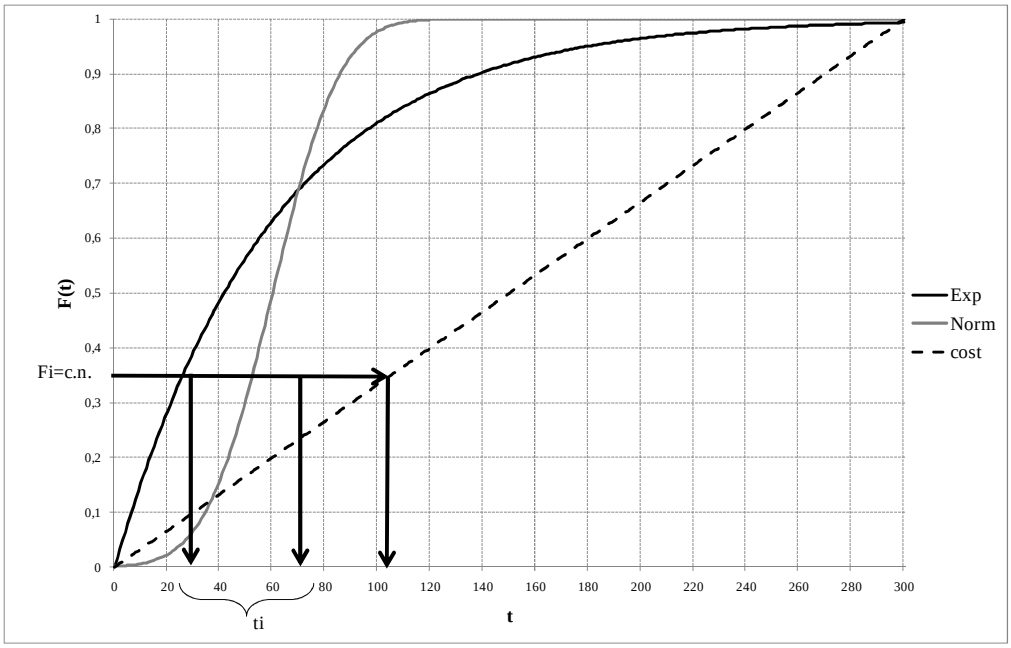


Figura 3.6: il motore di simulazione ad eventi discreti che determina il tempo di guasto in accordo cone la CDF degli eventi primari

Per esempio, per la stessa PAND di Figura 3.6, le *(E.3.9)* e *(E.3.10)* mostrano rispettivamente le condizioni logiche per il calcolo dell'affidabilità del sistema: S_{IE} è un valore booleano (vero/falso) che, ad ogni iterazione, indica se il sistema è funzionante oppure è guasto al tempo T_M. Se le condizioni logiche sono tali che S_{IE}=1 (il sistema funziona per tutta la durata del tempo di missione) il T_{IE} dalla PAND viene fatto corrispondere con il T_M, altrimenti corrisponde con il tempo di guasto del secondo ingresso T_{BEy}. Affinché la simulazione ad eventi discreti possa essere realizzata correttamente, tutti i tempi di guasto delle porte e degli input devono essere memorizzati per essere utilizzati nelle regole logico-temporali delle porte dinamiche dell'albero.

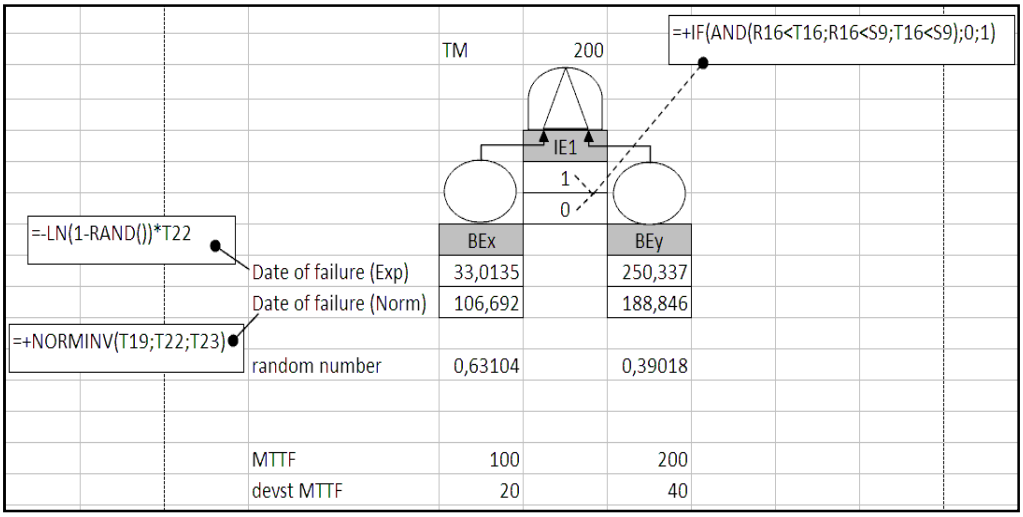


Figura 3.7 : implementazione della porta PAND a due ingressi in ambiente Excel®

$$S_{IE} = \begin{cases} 1 & \text{se TEx<H and TEx<TEy o TEx>H} \\ 0 & \text{se TEy<TEx<H} \end{cases} \qquad (E.3.9)$$

$$T_{IE} = \begin{cases} T_M & \text{se SIE=1} \\ TEy & \text{se SEI=0} \end{cases} \qquad (E.3.10)$$

Il principale vantaggio nell'uso dell'ambiente Excel® sta nel fatto che il motore simulativo viene implementato attraverso le sue funzioni standard; attraverso queste viene realizzato un linguaggio meta-strutturato che, diversamente dagli altri tool di affidabilità, può essere usato in qualsiasi computer vista la possibilità di integrazione di Excel® con i tanti software per l'ufficio come Open Office®, Google Docs®, ecc. Un ulteriore vantaggio dell'implementazione ad-hoc con Excel® è dato dal carattere "*wysiwyg*" dell'ambiente che offre la possibilità di leggere e comprendere le logiche delle porte semplicemente cliccando sulla cella che contiene la regola di funzionamento. Questa proprietà del foglio di calcolo non può che aumentare la robustezza e l'affidabilità di un software poiché, con la stessa logica dei software

open source, permette il controllo delle regole implementate e la loro estensione, favorendo la diffusione del know-how fra gli utilizzatori finali.

Infine, un ulteriore valore aggiunto dell'approccio risiede nel fatto che, mediante opportune modifiche del modello, è possibile implementare le logiche per la simulazione della disponibilità e del calcolo della prima occorrenza del TE per qualsiasi sistema, superando i problemi introdotti e non risolvibili con l'approccio gerarchizzato.

In Tabella 3.4 vengono mostrati i risultati e i rispettivi errori delle simulazioni per il sistema di Figura 3.4b già analizzato con la tecnica di gerarchizzazione: si può osservare come la simulazione permetta di ottenere dei risultati molto prossimi a quelli analitici.

Tabella 3.4: Inaffidabilità analitica per il DFT di Fig. 3.4(c) e calcolo tramite simulazione

λT_m	Unreliability	
	Analytic	**Simulation**
10	3.33×10^{-1}	3.35×10^{-1}
1	8.39×10^{-2}	8.38×10^{-2}
$\mathbf{10^{-1}}$	2.87×10^{-4}	2.96×10^{-4}
$\mathbf{10^{-2}}$	3.28×10^{-7}	3.31×10^{-7}
$\mathbf{10^{-3}}$	3.31×10^{-10}	3.37×10^{-10}

CONSIDERAZIONI SULL'USO DELL'APPROCCIO SIMULATIVO

Di seguito vengono riassunti i principali vantaggi e svantaggi della simulazione:

- allo stato dell'arte, è l'unico strumento che consente di modellare scenari per il calcolo della disponibilità della occorrenza di un evento indesiderato;
- può essere usata per effettuare dei confronti e controllare la robustezza dei risultati;
- in ambiente Excel® la proprietà *wysiwyg* migliora la condivisione del sapere, aumentando il senso di sicurezza e consapevolezza ;

- non vi sono limitazioni circa l'uso di distribuzioni di probabilità per gli input del sistema, grazie all'inversione delle leggi mediante estrazioni Monte Carlo;
- gli ambienti di calcolo possono essere facilmente integrati con i sistemi di controllo tipo DCS (Compagno & al., 2008), grazie ai quali diventa possibile alimentare il modello con dati reali in tempo reale. Infatti, nei moderni sistemi di controllo, gli ambienti delle applicazioni industriali (come per esempio impianti a rischio di incidente rilevante) sono fisicamente cablati mediante sensori, reti e bus di campo che, istante per istante, misurano e registrano lo stato di ogni dispositivo. Queste informazioni possono viaggiare a velocità elevatissime ed essere processate dai sistemi di controllo (DCS). All'interno del software del DCS possono facilmente essere implementati moduli software aggiuntivi, per cui l'utilizzo di un modello simulativo che fa uso delle informazioni del campo in tempo reale per la valutazione del rischio istantaneo è una possibilità ampiamente realizzabile e di grande interesse;
- per esigenze concrete è necessario adottare delle tecniche che consentano di velocizzare la convergenza della simulazione. Allo stato dell'arte, l'idea è quella di fare uso dei sistemi di calcolo distribuito (cloud computing) in modo da distribuire le routine di calcolo su un grande numero di macchine e processare le informazioni in parellelo. Basti pensare al costrutto "parfor" di Matlab® che, se lanciato su un'architettura parallela, implementa la modalità di calcolo distribuito automaticamente, schedulando i processi e ottimizzando le risorse.

Gli svantaggi dell'approccio simulativo sono vari. Infatti:

- nella valutazione di eventi riparabili, affinché le valutazioni siano congrue con quelle reali, bisogna scegliere appropriatamente il tempo di discretizzazione; infatti, l'evoluzione di ogni singolo evento (e input del sistema) viene sviluppata con sufficiente accuratezza soltanto se lo step di discretizzazione non è più ampio dei modi con cui si evolve un evento;
- a causa di questa prima osservazione, i tempi di simulazione possono diventare estremamente lunghi poiché più basso è lo step di discretizzazione, più lunghi si rivelano i tempi di calcolo;

- dovendo implementare l'ambiente di simulazione per ogni fault tree, a seconda della complessità del modello sono possibili errori di costruzione.

3.6 CASO STUDIO

Il caso studio più significativo è tratto dal Rapporto di sicurezza di una raffineria di petrolio. Si tratta di un modello di SFT (Figura 3.8) sviluppato dagli esperti di rischio della raffineria in seguito alle raccomandazioni che emergono dall'analisi di rischio HAZOP. Le valutazioni quantitative sono basate sulle informazioni statistiche relative agli eventi in ingresso all'SFT secondo i parametri della Tabella 3.5.

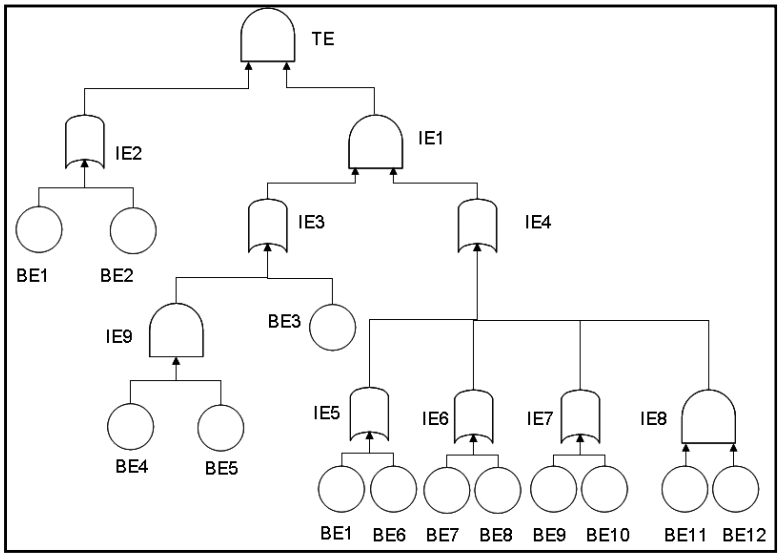


Figura 3.8: SFT di una sezione di un impianto di alchilazione per la raffinazione del petrolio

La Figura 3.9 mostra la versione dinamica dell'albero di Figura 3.8 Quest'ultimo è stato ottenuto cercando di implementare le logiche di sicurezza del sistema reale che nell'albero statico non possono essere modellate. Infatti, nella versione statica le porte IE1, IE8 e la TOP sono rappresentate dalle tradizionali AND. Ciò comporta una descrizione non esaustiva della logica di sicurezza delle pompe secondo cui la G17 è la pompa primaria e la G17S è in stand by freddo, fungendo da ricambio on-line per la G17. Nel DFT, la porta IE8 viene quindi sostituita da una SPARE al fine di modellare una configurazione stand-by freddo delle pompe dell'impianto. Due

porte PAND sostituiscono la IE1 e la TOP, in modo da considerare le priorità tempo dipendenti che i sistemi di sicurezza (IE2 e IE3, che comprendono allarmi, sistemi di blocco e di intervento operativo dell'uomo) hanno per i guasti delle sezioni di impianto che monitorano (IE1, IE4).

Tabella 3.5: Parametri degli Eventi Base (BEs) per il FT (λ–tasso di guasto; q-probabilità costante)

ID	Description	λ	q
		[h^{-1}]	[-]
BE1	Errore Umano	-	1.0×10^{-3}
BE2	Guasto HV72	9.1×10^{-4}	-
BE3	Mancato Intervento Operativo	-	1.0×10^{-5}
BE4	Guasto LAHH78	1.7×10^{-4}	-
BE5	Guasto LAHH	7.5×10^{-4}	-
BE6	Guasto HV75	9.1×10^{-4}	-
BE7	Guasto Flussostato	4.5×10^{-3}	-
BE8	Guasto FV72	8.6×10^{-4}	-
BE9	Guasto Controllo di Livello	4.5×10^{-4}	-
BE10	Guasto LAHH78	7.9×10^{-3}	-
BE11	Guasto Pompa G17	1.5×10^{-4}	-
BE12	Guasto Pompa G17S	9.5×10^{-4}	-

Considerando i parametri della Tabella 3.5, la risoluzione analitica del DFT di Figura 3.9 diviene impraticabile.

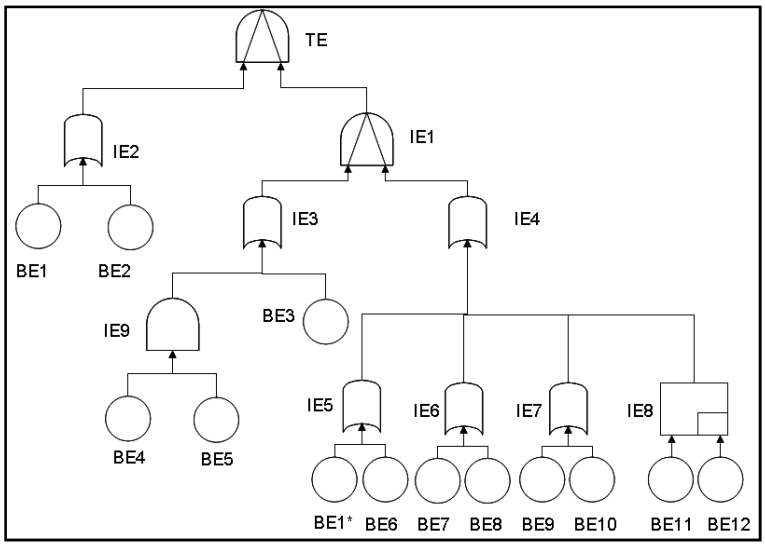


Figura 3.9: DFT dell'impianto di alchilazione

Infatti, il modello è caratterizzato da:

- eventi di probabilità costante (BE1 e BE3) che invalidano l'uso delle CTMC;
- un evento ripetuto (BE1) al livello più basso del DFT che non verifica le ipotesi per l'utilizzo della gerarchizzazione.

Per questi motivi, l'approccio tramite la simulazione ad eventi discreti risulta l'unico praticabile. Al fine di validare i risultati deducibili attraverso la simulazione, sono state apportate delle modifiche al DFT originale in modo da rendere possibile la risoluzione analitica e confrontare questi risultati con quella dell'ambiente di simulazione. Infatti, come detto nel paragrafo che descrive la tecnica di simulazione ad eventi discreti, uno dei vantaggi di questo approccio sta nel fatto che il modello ad eventi del DFT, una volta costruito, rimane inalterato, mentre le proprietà (in termini di CDF) degli eventi possono essere variate senza intaccare la correttezza del modello.
I modelli relativi agli scenari ipotizzati (chiamati Test N°) sono stati costruiti tenendo conto della classificazione in Figura 2.14:
Test 1: tutti i BE sono caratterizzati da una distribuzione esponenziale negativa, per cui il BE1 e il BE3 vengono settati con un tasso di guasto equivalente calcolato mediante l'equazione *(E.3.6)*;
Test 2: questo test rappresenta la configurazione originale degli ingressi in Tabella 4.5 proveniente dal caso reale;
Test 3: BE1 e BE3 vengono nuovamente descritti attraverso una distribuzione esponenziale negativa (come nel Test 1); gli altri BE sono caratterizzati da una distribuzione di Weibull con un parametro di scala uguale al tasso di guasto λ_i (dei rispettivi BE) ed un fattore di forma pari a 3.

Inoltre, per l'albero UnMOE, l'evento ripetuto BE1 viene sostituito con un evento BE1* equivalente tale che BE1 e BE1* presentino le stesse caratteristiche aleatorie pur essendo due eventi del tutto distinti.
I risultati dei modelli SFT e DFT sono mostrati nelle Tabelle 3.6 e 3.7 dove:

- "nf" è l'acronimo di "not feasible", indica che il software non è stato in grado di risolvere il modello;

- "np", acronimo di "not performed", indica che il modello non è stato risolto poiché avrebbe richiesto degli studi di fattibilità che vanno oltre gli obiettivi di questo lavoro;
- "-" indica che il software non ha fornito dei risultati finali poiché, nell'esecuzione, il computer ha raggiunto una condizione di overflow.

Per lo SFT (risultati in Tabella 3.6) il calcolo dell'affidabilità viene risolto senza problematiche da parte di tutti i tool. L'unico problema si presenta per quanto concerne il Test 3, in ambiente Relex®, dal momento che questo software non ammette ingressi caratterizzati da distribuzione di Weibull; per risolvere lo SFT, dunque, bisogna calcolare a parte le probabilità di guasto di ogni BE caratterizzato da una distribuzione generalizzata al tempo T_M di missione e inserirlo come se fosse una probabilità costante nel modello. Per lo SFT di tipo MOE, come già ribadito, il software Galileo non può fornire alcun risultato vista l'incapacità di processare eventi ripetuti.
Analizzando i risultati ottenuti, è possibile notare come l'affidabilità dello SFT nei casi MOE e UnMOE non variano significativamente.
Per quanto riguarda il modello dinamico di DFT (Tabella 3.7), si osserva che:

1. non può essere risolto con SHARPE (per la mancanza del DFT-wizard);
2. i modelli di DFT con il MOE sono risolti soltanto mediante l'approccio simulativo poiché Galileo non implementa i MOE mentre, per il caso studio in questione, il software Relex® si blocca durante l'esecuzione (probabilmente in quanto il DFT monolitico incorre nel problema dell'esplosione nello spazio degli stati, generando una CTMC con circa 2^{12} stati).

Per il DFT UnMOE, in configurazione monolitica si è osservato che Relex® presenta lo stesso problema di overflow del caso MOE, mentre Galileo e la simulazione restituiscono risultati paragonabili. Per quanto riguarda i tempi di questi ultimi due tool di risoluzione eseguiti con un pc portatile (Intel® core-duo da 1.86GHz e 2 Gb di Ram), il Galileo ha impiegato circa 30 minuti mentre la simulazione, con un numero di iterazioni pari 10^8, ha impiegato circa 9 ore.

Tabella 3.6: Risultati per il modello statico di FT relativo all'impianto di alchilazione

		Scenari Ipotizzati		
		Test1	Test2	Test3
MOE	Relex®	$7.73x10^{-1}$	$7.73x10^{-1}$	$9.48x10^{-1}$
(BE1=BE1*)	Galileo	n.f.	n.f.	n.f.
	SHARPE	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.48x10^{-1}$
	Simulative approach	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.42x10^{-1}$
UnMOE	Relex	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.63x10^{-1}$
(BE1≠BE1*)	Galileo	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.63x10^{-1}$
	SHARPE	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.63x10^{-1}$
	Simulative approach	$7.\ 73x10^{-1}$	$7.73x10^{-1}$	$9.50x10^{-1}$

Tabella 3.7: Risultati per il modello dinamico di FT relativo all'impianto di alchilazione

		Scenari Ipotizzati		
		Test1	Test2	Test3
MOE	Relex®	-	n.f.	n.f.
(BE1=BE1*)	Galileo	n.f.	n.f.	n.f.
	SHARPE	n.f.	n.f.	n.f.
	Simulative approach	$5.35x10^{-5}$	$5.49x10^{-5}$	$<10^{-7}$
UnMOE	Relex® Monolithic	-	n.f.	n.f.
(BE1≠BE1*)	Relex® Strong Hierarchy	$5.32x10^{-5}$	n.f.	n.f.
	Relex Weak Hierarchy	$4.47x10^{-4}$	n.f.	n.f.
	Galileo Monolithic	$5.31x10^{-5}$	n.f.	$7.93x10^{-9}$
	Galileo Strong Hierarchy	$5.32x10^{-5}$	n.f.	n.p.
	Galileo Weak Hierarchy	$4.47x10^{-4}$	n.f.	n.p.
	SHARPE	n.f.	n.f.	n.f.
	Simulative Approach	$5.08x10^{-5}$	$5.07x10^{-5}$	$<6.00x10^{-8}$

La risoluzione mediante l'approccio gerarchico (esatto e non) è stata implementata solo per il Test 1, così da comprendere quali sono gli ordini delle approssimazioni introdotte. Si fa notare che grazie all'approccio entrambi i tool analitici di Relex® e Galileo sono in grado di risolvere il modello equivalente di DFT*, restituendo degli identici risultati. Il modello mediante l'approccio esatto viene riportato in Figura 3.9a, quello inesatto viene condensato attraverso un passaggio in più dell'algoritmo di gerarchizzazione, in Figura 3.9b.

Per rendere possibile l'applicabilità dell'algoritmo di gerarchizzazione, il DFT originale è stato prima modificato attraverso ulteriori semplificazioni sul modello, motivate dalle seguenti considerazioni tecniche in una fase di pre-analisi:

- la porta IE8 viene eliminata poiché il suo contributo al guasto della porta IE4 è minimo (dell'ordine di 10^{-8});

- l'evento BE3 viene rimosso dal momento che il suo tasso di guasto equivalente è basso rispetto al tasso di guasto equivalente della porta IE9.

Attraverso queste considerazioni il modello di DFT* di Figura 3.9a, gerarchizzato con l'approccio esatto, risulta formato da un evento equivalente $BE_{1\text{-}2}$ che sostituisce il sottosistema composto al di sotto della porta IE2 e dal sottosistema IE1.

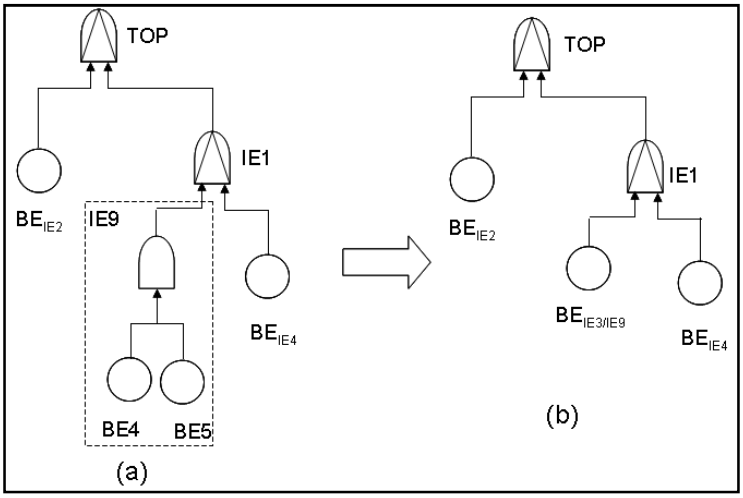


Figura 3.9: Approccio esatto (a) e non esatto (b) per la gerarchizzazione del modello della raffineria

Il sottosistema composto IE1 di Figura 3.9a (il blocco che comprende tutti gli elementi al di sotto della PAND IE1), in particolare, presenta la stessa struttura dell'albero analizzato in Figura 3.4b; infatti, la porta IE4 (privata della porta IE8) viene rimpiazzata da un evento equivalente BE_{IE4} attraverso una gerarchizzazione esatta (che garantisce l'equivalenza del sistema composto con quello originale) caratterizzata da un tasso di guasto equivalente pari alla somma dei tassi di guasto dei singoli BE_i al di sotto della porta IE4, come segue:

$$\lambda_{IE4} = \lambda_{BE1^*} + \lambda_{BE6} + \lambda_{BE7} + \lambda_{BE8} + \lambda_{BE9} + \lambda_{BE10} = 2.235 \times 10^{-2}$$

La configurazione di Figura 3.9b, invece, è ottenuta attraverso un'altra gerarchizzazione che coinvolge la porta IE9. Si tratta di una gerarchizzazione

inesatta sulla porta AND IE9, mediante la composizione dell'evento $BE_{IE3/IE9}$ attraverso l'equazione *(E.3.8)* per il calcolo del tasso equivalente.

$$\lambda_{IE3/IE9} = \frac{1}{MTTF_{IE9}} = \frac{\lambda_{BE4}^2 \lambda_{BE5} + \lambda_{BE5}^2 \lambda_{BE4}}{\lambda_{BE4}^2 + \lambda_{BE5}^2 + \lambda_{BE4}\lambda_{BE5}} = 1.6316 \times 10^{-4}$$

Una volta realizzati tali modelli gerarchizzati di DFT*, possono essere riproposte le stesse considerazioni viste per gli esempi Figura 3.4a e Figura 3.4c. Come mostrato nella Tabella 3.7, i risultati dell'approccio gerarchico esatto sono molto prossimi a quelli che si ottengono mediante la risoluzione analitica del modello monolitico di DFT (risolti attraverso Galileo e la simulazione). La gerarchizzazione inesatta, invece, ha introdotto un errore relativo di circa il 740% che è affine all'ordine di errore percentuale (nel range di $10^2 \div 10^3$, vedi Tabella 3.2) calcolato nella stessa configurazione dell'esempio di Figura 3.4c, in relazione al parametro caratteristico λt. Il confronto può essere ottenuto considerando i risultati di Tabella 3.2 e quelli della Tabella 3.7 considerando che per quest'ultima il parametro caratteristico λt risulta (al tempo T_M =8760h) dell'ordine di 10^{-1} (dato dal prodotto tra il T_M e il tasso di guasto tipico dei BE).

Gli scenari dei Test 2 e 3 sono stati risolti soltanto con la simulazione. Si è notato come la distribuzione di Weibull stravolge i risultati di entrambi gli alberi di guasto statico e dinamico. Mentre per il primo l'inaffidabilità del sistema cresce, per il DFT questa viene drasticamente ridotta a valori (il cui ordine di grandezza è compreso tra $10^{-9} \div 10^{-8}$) che possono essere considerati trascurabili nell'ambito della sicurezza degli impianti industriali. Questo risultato è interessante perché può servire a riflettere sulla robustezza delle ipotesi su cui si basano i modelli di rischio. Ciò dimostra che la teoria della dependability ha messo in piedi delle tecniche di valutazione formalmente corrette le cui ipotesi di partenza, però, meritano di essere approfondite. Per questo motivo è conveniente proporre, quando realizzabile, la tecnica di simulazione come importante benchmark per le valutazioni dei rischi associati ai comportamenti dei sistemi complessi.

Nelle Figure 3.10, 3.11, 3.12 e 3.13 sono proposti i risultati delle simulazioni dei Test 1 e Test 2 (due simulazioni per ogni test) eseguite con 10^7 iterazioni e i relativi intervalli di confidenza, rispetto ad un livello di significatività α di 0.01.

Si può notare (Figure 3.10 e 3.11) che nel Test 1 occorrano circa $6x10^6$ ricalcoli per ogni simulazione per ottenere una convergenza del risultato e un risultato di confidenza decisamente incoraggiante (compreso tra $10^{-9} \div 10^{-7}$).

Per il Test 2 (Figure 3.12 e 3.13) l'assestamento del risultato in un intervallo di confidenza (dell'ordine di 10^{-10}) si ha con $8x10^6$ ricalcoli circa.

QSFT = inaffidabilità SFT (simulazione);
QDFT = inaffidabilità DFT (simulazione);
QDFT-Galileo = inaffidabilità del DFT (Galileo);
α = livello di significatività

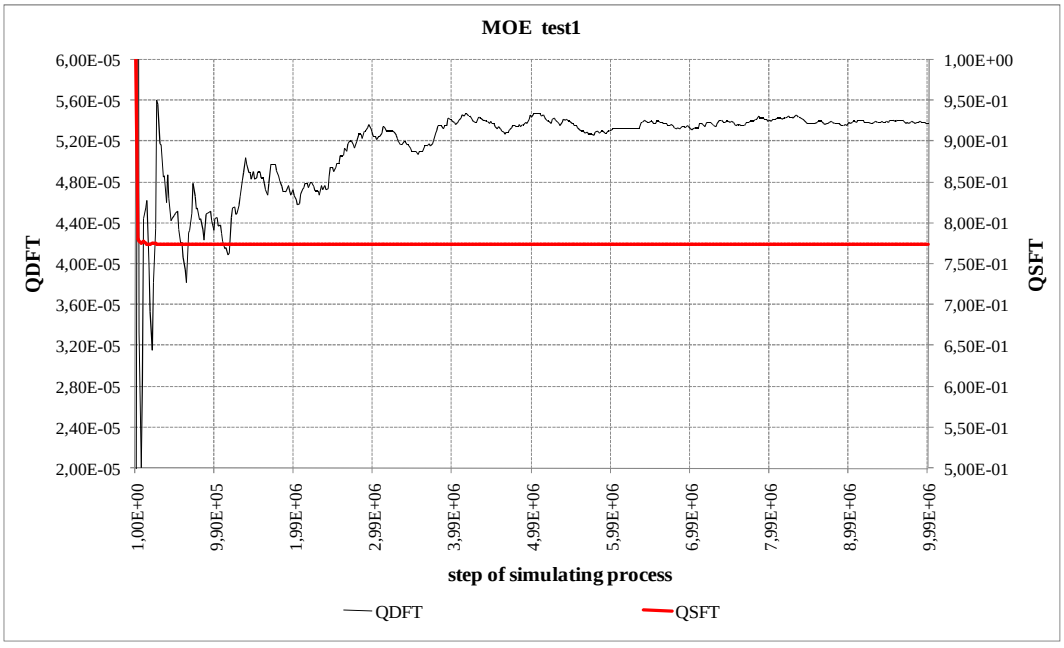


Figura 3.10: Risultati Approccio Simulato, MOE (Test1) (α=0.01; confidenza=$1.85x10^{-9}$)

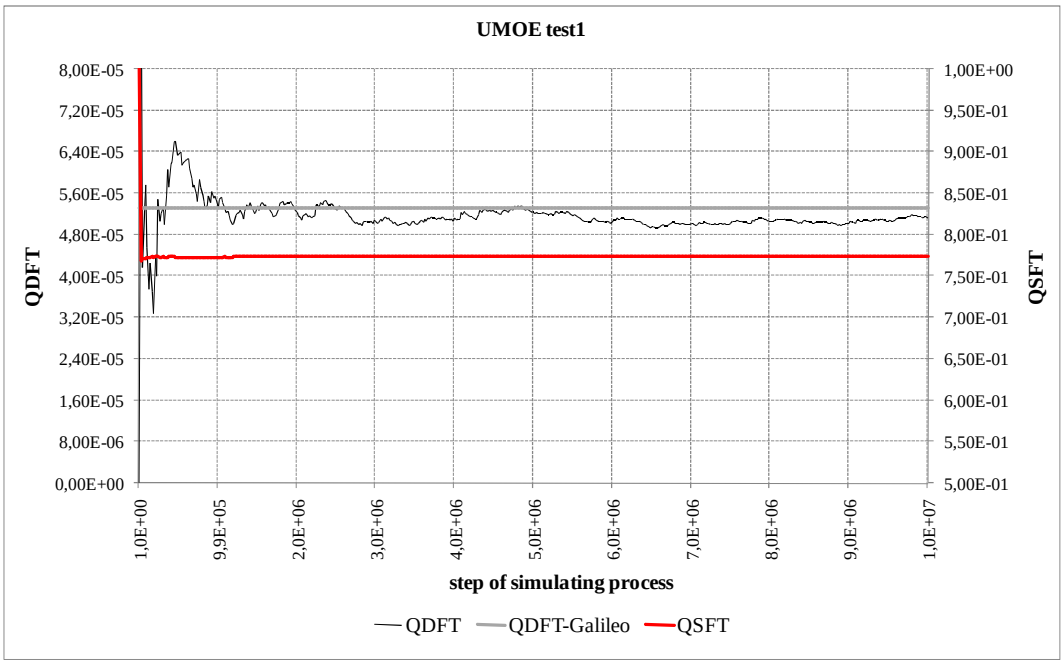


Figura 3.11: Risultati Approccio Simulato, MOE (Test1) (α=0.01; confidenza=4.91x10^{-7})

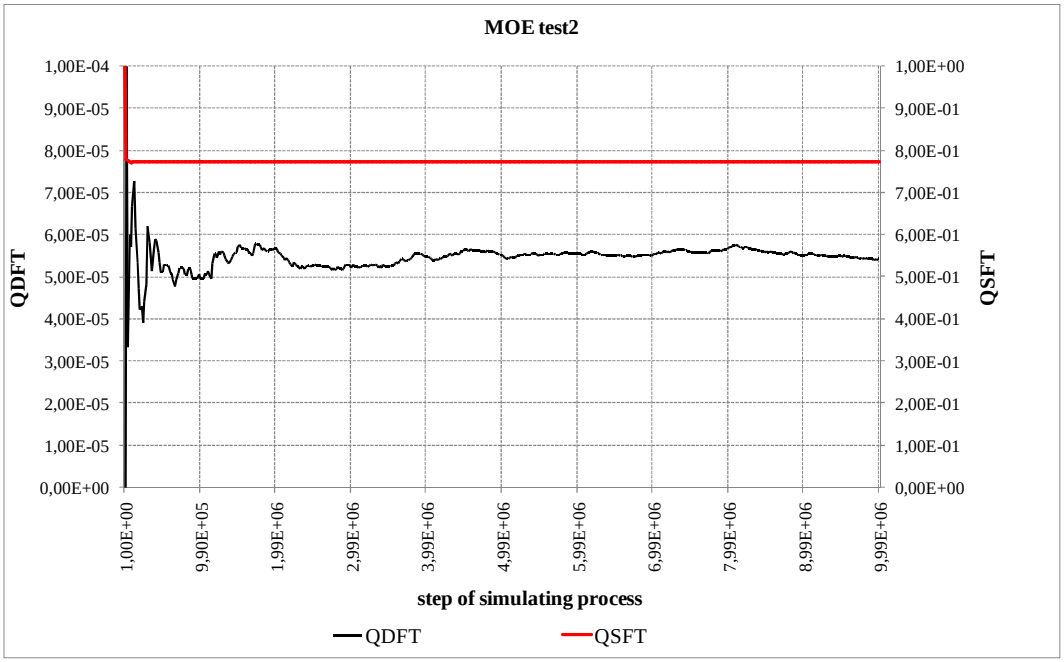


Figura 3.12: Risultati Approccio Simulato, MOE (Test2) (α=0.01; confidenza=9.62x10^{-10})

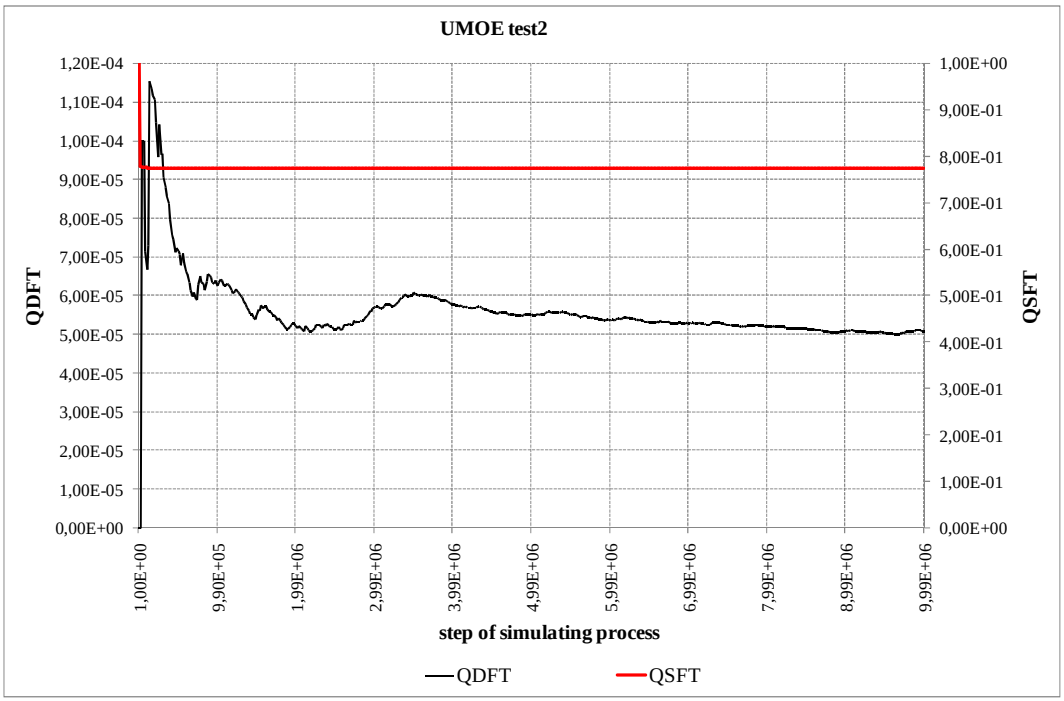


Figura 3.13: Risultati Approccio Simulato, MOE (Test2) (α=0.01; confidenza=5.35x10^{-10})

Per il calcolo delle IMs del caso statico si è fatto uso del software SHARPE mentre, per il caso dinamico (sotto opportune ipotesi), si sono utilizzati il software Relex® e la simulazione ad eventi discreti, facendo uso dell'approccio diretto al calcolo della BIM (FTM) secondo l'equazione *(E.2.9)*, che riportiamo:

$$BIM^{X_i}(t) = R_{sys}(1_i, X) - R_{sys}(0_i, X) = F_{sys}(0_i, X) - F_{sys}(1_i, X)$$

Questo approccio è giustificato poiché il DFT presenta una struttura coerente. Siccome non possiamo essere certi dell'equivalenza tra BIM ed FTM, nel caso dinamico indicheremo queste misure con l'acronimo FTM.

I risultati delle IMs nel caso statico (BIM) sono riportati in Figura 3.14 e mostrano l'andamento di queste grandezze rispetto al tempo di missione.

Da questi grafici si possono fare alcune importanti osservazioni, utili per il modello DFT:

1. tutti gli eventi che afferiscono alla porta IE4 hanno, fin dai primi istanti del tempo di missione, una BIM nulla (non sono visibili poiché schiacciati sull'asse dei tempi).
2. rispetto alla BIM, gli eventi a probabilità costante (BE_1 e BE_3) nel tempo perdono di significatività rispetto ai guasti dei componenti BE_2, BE_4 e BE_5.

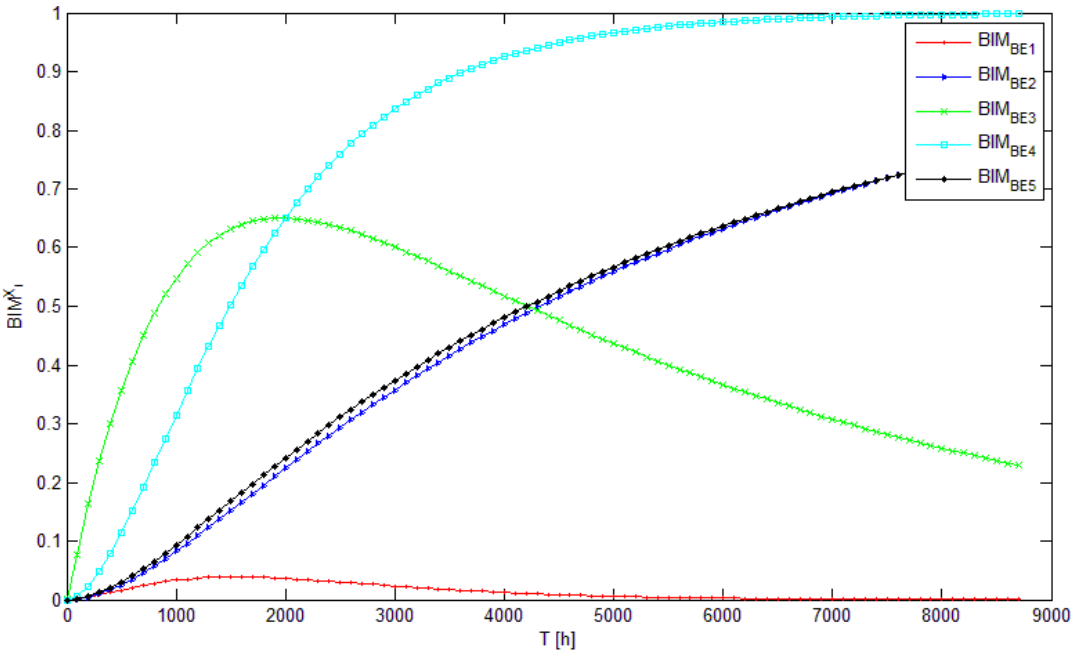


Figura 3.14: Andamento nel tempo della Birnbaum Importance Measure per lo SFT della raffineria. I BE_6, BE_7, BE_8, BE_9, BE_{10}, BE_{11} e BE_{12} sono insignificanti secondo questa metrica

Per il DFT, l'idea è quella di utilizzare i dati dello scenario del Test1 (tutti gli eventi BE caratterizzati da una distribuzione negativa esponenziale) con il modello in Figura 3.15 che presenta una gerarchizzazione esatta rispetto al sottomodello della porta IE4, vincolata all'ipotesi (vista in precedenza) di eliminare gli eventi BE_{11} e BE_{12} il cui contributo all'affidabilità del sistema è trascurabile.

In questo modo, il software Relex® non soffre del problema di overflow legato alla conversione nello spazio degli stati.

Per la simulazione, invece, sarà necessario sviluppare 12 simulazioni (piuttosto che le 26 del DFT originale), poiché 6 sono i BE in ingresso al modello e ognuno di loro richiede 2 simulazioni (per il calcolo diretto della FTM). Inoltre, utilizzando questo approccio, il calcolo di ogni IMs viene fissato ad un istante ben preciso (il

T_M=8760h), per cui non si potrà sviluppare l'andamento nel tempo come fatto nel caso SFT (Figura 3.14).

In Tabella 3.8 e 3.9 sono riportati i risultati della FTM e della CFT, calcolata mediante l'approccio diretto al tempo T_M = 8760h, con Relex® e la simulazione (in $1x10^6$ ricalcoli), mostrando la bontà della precisione raggiunta.

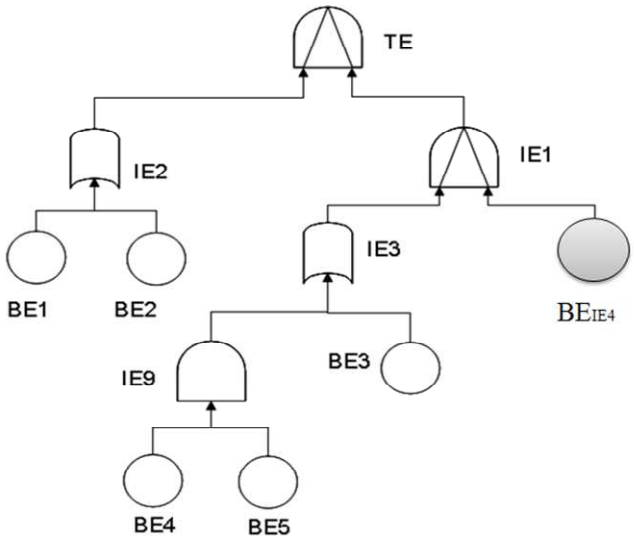


Figura 3.15: Rimodellazione gerarchizzata del DFT della raffineria. Il sottomodello composto della porta IE4 utilizza una gerarchia esatta

Tabella 3.8: FTM del modello in Figura 3.15 calcolata attraverso Relex® e la simulazione usando l'approccio diretto

FTM	BE_1	BE_2	BE_3	BE_4	BE_5	BE_{IE4}
Sim	$1.20x10^{-2}$	$3.00\ x10^{-2}$	$2.27\ x10^{-1}$	$1.14x10^{-1}$	$3.40x10^{-2}$	0
Relex®	$1.24\ x10^{-2}$	$3.04\ x10^{-2}$	$2.27x10^{-1}$	$1.14x10^{-1}$	$3.42x10^{-2}$	0

Si può notare come nel caso dinamico la FTM del BE_3 sia la più elevata fra le FTM, diversamente da quanto accadeva nel caso statico.

La misura CFT ridimensiona l'importanza del BE_3 e come ci si poteva aspettare ritorna evidente la criticità dei BE_2, BE_4 e BE_5. Gli elevati valori della CFT dei BE_2, BE_4 e BE_5 sono dovuti al rapporto (vedi *(E2.10)*) tra il valore dell'inaffidabilità del componente e quello del sistema al tempo T_M (si ricorda che nel caso dinamico il valore dell'inaffidabilità del sistema è dell'ordine di 10^{-5}, per cui per i BE_1 e BE_5 questo si mantiene basso diversamente che per gli altri BE).

Tabella 4.9: CFT del modello in Figura 3.15 calcolata con i dati della FTM

CFT	BE_1	BE_2	BE_3	BE_4	BE_5	BE_{IE4}
	2.22×10^{-3}	5.6×10^{2}	4.1×10^{-4}	1.65×10^{3}	6.35×10^{1}	0

È interessante fare notare che il sottomodello composto della porta IE4 non sia rilevante secondo le misure di importanza. Questa informazione è molto importante dal punto di vista della PRA. Infatti, se da un lato il risultato è discutibile (in quanto sembra non attribuire alcun valore agli elementi che compongono quella sezione del modello) dall'altro fornisce due indicazioni che meritano attenzione.

La prima è di ambito manutentivo: infatti, sebbene sia vero che la FTM_{IE4} è nulla, è altrettanto vero che se nessuno degli elementi del sottosistema IE4 si guastasse il sistema sarebbe completamente affidabile. Questo ci indica che un aumento dell'affidabilità del sistema si ottiene se si agisce non esclusivamente su uno solo degli elementi di IE4 (d'altra parte sono tutti in logica OR), quanto piuttosto cercando di migliorare tutti gli elementi di IE4 o agendo sulla logica della struttura.

La seconda informazione è di carattere modellistico. Infatti, la FTM_{IE4} risulta nulla perché il contributo al guasto del sistema apportato dalla porta IE4 (sia essa guasta o funzionante) è sempre lo stesso (infatti, la FTM valuta il sistema nei casi estremi secondo la *(E.2.9)*, ponendo la IE4 guasta o perfettamente funzionante).

Seguendo lo schema logico del DFT e risolvendolo a vista si ha che:

- ponendo la IE4 sempre affidabile, il sistema non perviene al TE perché è come se nessuno degli eventi che afferiscono alla IE4 si guasta non consentendo il guasto della PAND;
- nel secondo caso, ponendo IE4 (che è il secondo ingresso alla PAND IE1) sempre guasto (impostando un tasso di guasto infinitamente grande), la logica della PAND suggerisce che il sistema venga dirottato verso uno stato "safe", che non verifica le condizioni di guasto della PAND.

Questa seconda osservazione, deve far riflettere sulla convenienza dell'uso della PAND nelle logiche puramente affidabilistiche. Quando non vengono considerate le transizioni di riparazione che riportano il sistema allo stato originale (non lasciandolo

nello stallo di uno stato assorbente di tipo "safe"), questa logica può dunque risultare priva di rigore (in appendice è sviluppato il caso della porta PAND a due ingressi e spiegato con più dettaglio il problema qui evidenziato, risolto nella logica di un suo utilizzo nell'ambito della disponibilità o del calcolo della prima occorrenza di un evento di guasto).

Al fine di chiarire ulteriormente alcuni aspetti relativi alla valutazione delle IMs nei sistemi dipendenti, si riporta un secondo caso studio tratto da (Kim et al., 2010 in publishing), riguardante una Sensor Network. I parametri della Sensor Network sono riportati in Tabella 3.10.
L'analisi di affidabilità viene condotta mediante un modello dinamico ibrido SFT-CTMC appartenente alla classe dei sistemi gerarchizzati in maniera esatta (come in Figura 3.3), dove i sottosistemi composti (input dello SFT) ai livelli bassi dell'albero sono CTMC piuttosto che DFT (Figura 3.16).

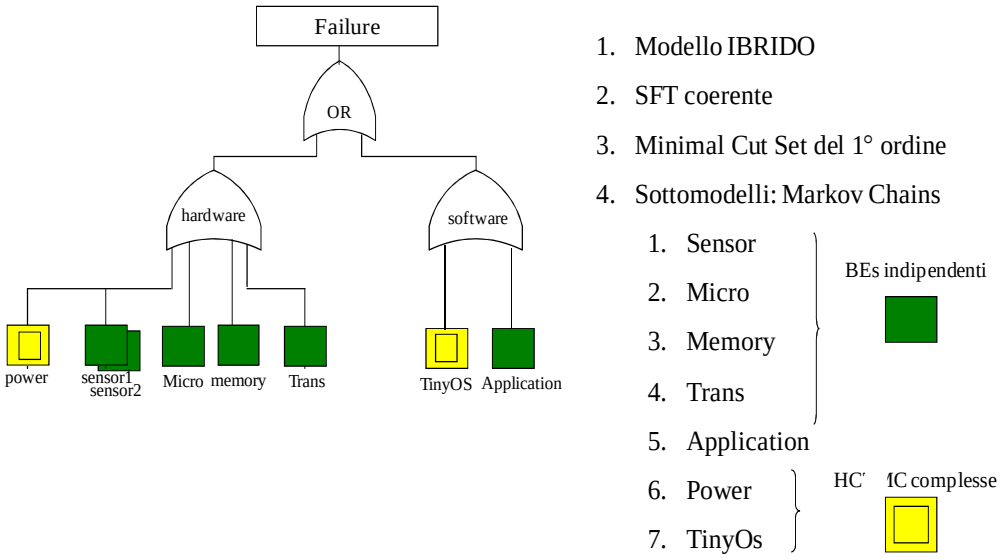


Figura 3.16: modello ibrido SFT con CTMC al livelli più bassi dell'albero

Tabella 3.10: Parametri della Sensor Network (f – frequenza di guasto o riparazione o accadimento; p - probabilità costante)

ID	Descrizione	F	p
		$[h^{-1}]$	
β_{wkp}	Tasso di risveglio	1.0×10^{-2}	
α_{slp}	Tasso di riposo	1.0×10^{-8}	
λ_{bd}	Tasso di scarica batteria	1.41×10^{-4}	
λ_{c}	Tasso di guasto componenti	1.0×10^{-4}	
λ_{pwr}	Tasso di guasto permanente Power	1.0×10^{-4}	
λ_{app}	Tasso di guasto livello Application	1.0×10^{-5}	
λ_{v}	Tasso vulnerabilità del TinyOS	1.0×10^{-2}	
λ_{fp}	Tasso di possibile guasto TinyOS	1.4×10^{-3}	
λ_{f1}	Tasso di guasto TinyOS	2.0×10^{-3}	
λ_{uc}	Tasso di rivelazione attaccoTinyOS	6.0×10^{0}	
λ_{a}	Tasso di attacco al TinyOS	4.1×10^{-2}	
μ_{s}	Tasso di riparazione del TinyOS	6.0×10^{0}	
μ_{rj}	Tasso di Rejuvenation del TinyOS	2.0×10^{1}	
C_s	Probabilita di successo della rilevazione attacco al TinyOS	-	6.0×10^{-1}
C_{rj}	Probability di successo della Rejuvenation del TinyOS	-	9.5×10^{-1}

In particolare, i sottosistemi (vedi Figura 3.17 e Figura 3.18):

- Sensor, Micro, Memory e Transceiver (trans) sono i componenti fisici della Sensor Network e vengono classificati con la dicitura "Components". Da un punto di vista affidabilistico, sono modellati mediante una CTMC a due soli stati (funzionamento e guasto) con un medesimo tasso di guasto λ_c
- l'Application, è lo strato software di livello più alto nello stack ISO/OSI del protocollo TCP/IP semplificato per la Sensor Network. Da un punto di vista affidabilistico tutti i servizi offerti dal livello Application sono modellati mediante una CTMC a due soli stati (funzionamento e guasto) con un tasso di guasto pari a λ_{app};
- il "Power" è il sistema di alimentazione. Da un punto di vista affidabilistico viene descritto da una CTMC a più stati in cui sono previsti regimi di funzionamento in modalità normale e modalità di attesa (sleep), nonché guasti di natura interna e da esaurimento della batteria;

• il "TinyOS" rappresenta il sistema operativo su cui si basa la Sensor Network. Il comportamento affidabilistico di questo sistema software viene modellato mediante una CTMC abbastanza articolata (Figura 3.18) in cui vengono previsti guasti dovuti ad attacchi hacker o bug informatici, nonché meccanismi di ripristino del normale funzionamento del sistema. Trattandosi di applicazioni informatiche, è normale avere delle frequenze di accadimento elevate (come il numero di attacchi o i tentativi di rilevazione dei sistemi di sicurezza del sistema operativo, ecc.).

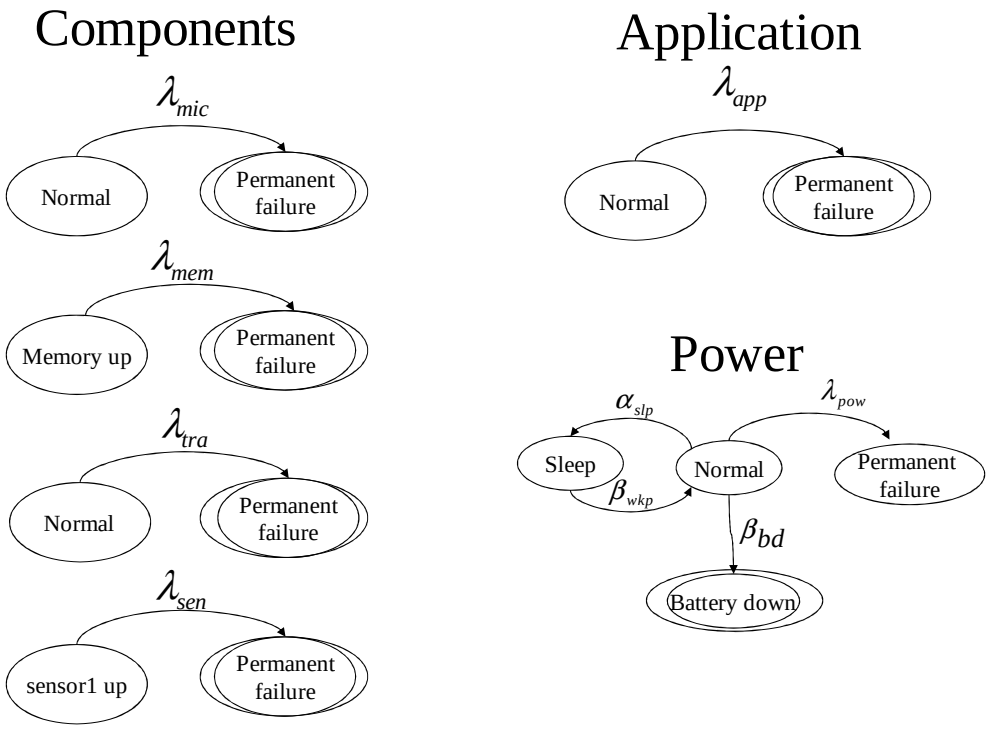


Figura 3.17: Rappresentazione delle CTMC dei sottomodelli della Sensor Network

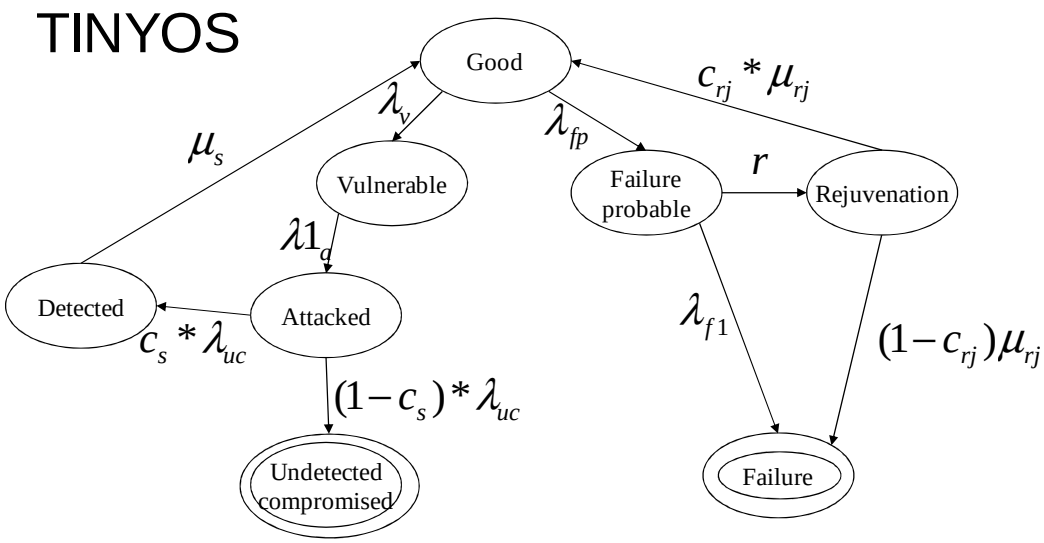


Figura 3.18: sottosistema TinyOS della Sensor Network

La logica di affidabilità del sistema è molto semplice poiché sia un guasto hardware e sia un guasto software causano un'interruzione del servizio della rete.

Una volta risolti le CTMC, lo SFT è alimentato istante per istante con i valori dell'inaffidabilità di questi sottosistemi. Trattandosi di un SFT, anche il calcolo dell'inaffidabilità è altrettanto semplice e può essere effettuato mediante l'equazione (*A.9*) in appendice, l'unione delle inaffidabilità dei singoli sottosistemi. L'andamento dell'inaffidabilità del sistema rispetto al tempo è mostrato in Figura 3.19.

Si osserva che il contributo più elevato all'inaffidabilità del sistema è dato dal TinyOS seguito dal Power e dai Components (i componenti fisici del sistema) L'inaffidabilità del componente Application è relativamente trascurabile rispetto alle altre.

Attraverso le IMs si possono ottenere ulteriori informazioni, al fine di validare le considerazioni appena espresse. In SHARPE è possibile istruire un programma (vedi appendice) (Sahner et al., 1996), (Sahner & Trivedi, 1987), (Pan & Trivedi, 2001) che esegua queste routine utilizzando le funzioni primitive ***bimpt*** (che restituisce la BIM), ***simpt*** (che restituisce la SIM) e ***cimpt*** (che restituisce la CIF).

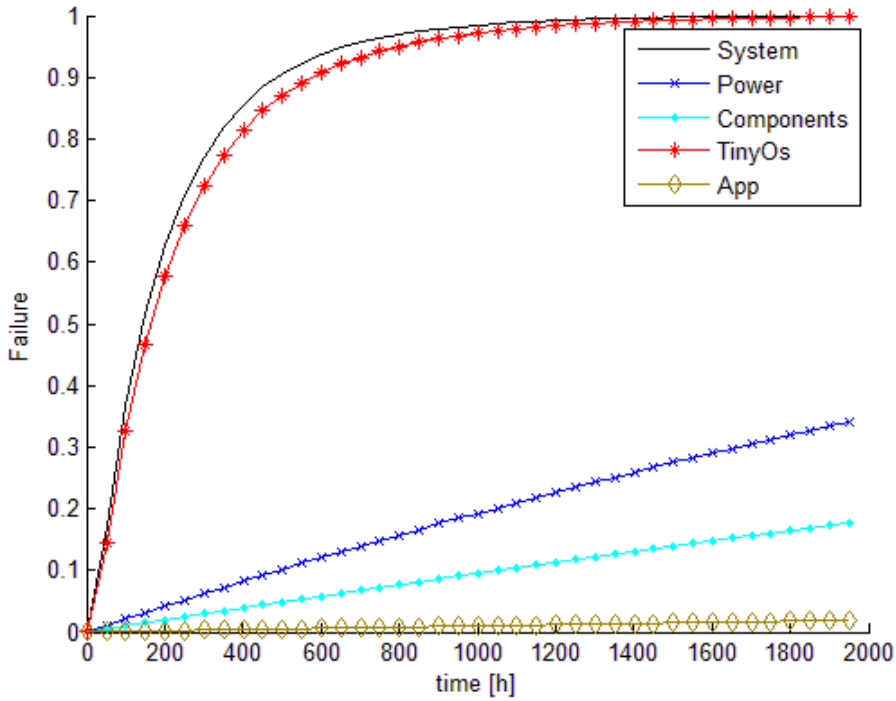


Figura 3.19: Inaffidabilità del Sistema e delle sue parti. Il TinyOS è il sottosistema che contribuisce maggiormente al guasto della Sensor Network

Il valore della SIM è identico per tutti i componenti e questo è un risultato atteso dal momento che il modello presenta solamente delle porte OR, tale per cui ogni evento ha la stessa criticità sotto il punto di vista della struttura.

In Figura 3.20 è riportato l'andamento della BIM. In questo caso la BIM corrisponde esattamente con la FTM poiché, una volta risolti i sottomodelli composti, il modello di partenza può essere trattato come uno SFT tradizionale.

Per quanto riguarda l'attività di classificazione per importanza degli input dello SFT, si può osservare che il TinyOS è l'elemento più critico mentre la valutazione sul sottocomponente Power è apprezzabile solamente attraverso la valutazione della CIF (Figura 3.21 e dettaglio di ingrandimento): il Power risulta più critico rispetto agli altri elementi soltanto durante le prime ore del tempo di missione, per poi stabilizzarsi a dei valori paragonabili con quelli degli altri elementi.

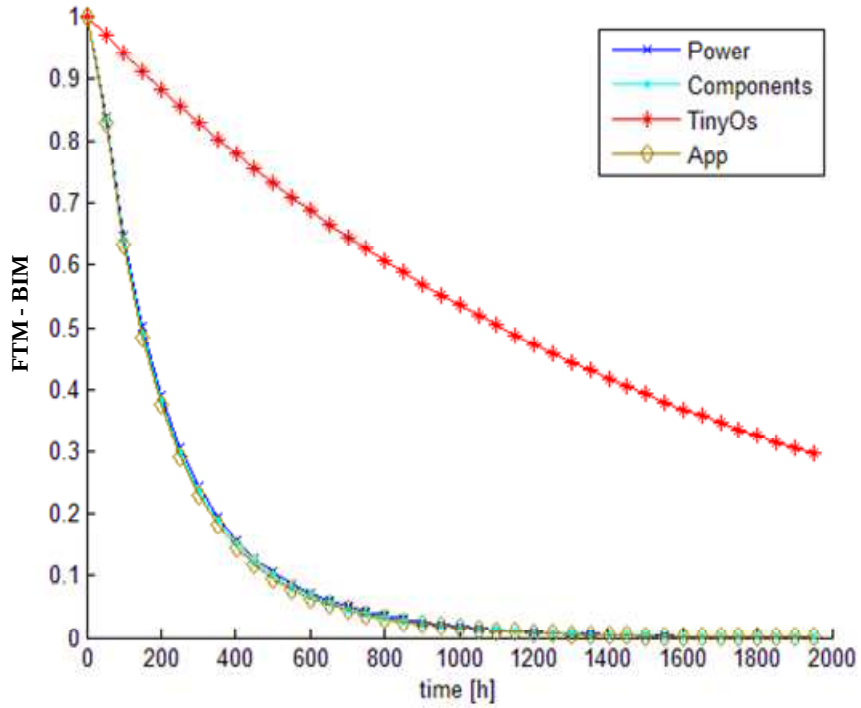


Figura 3.20: FTM dei dispositivi della Sensor Network. In questo caso, la FTM corrisponde esattamente con la BIM.

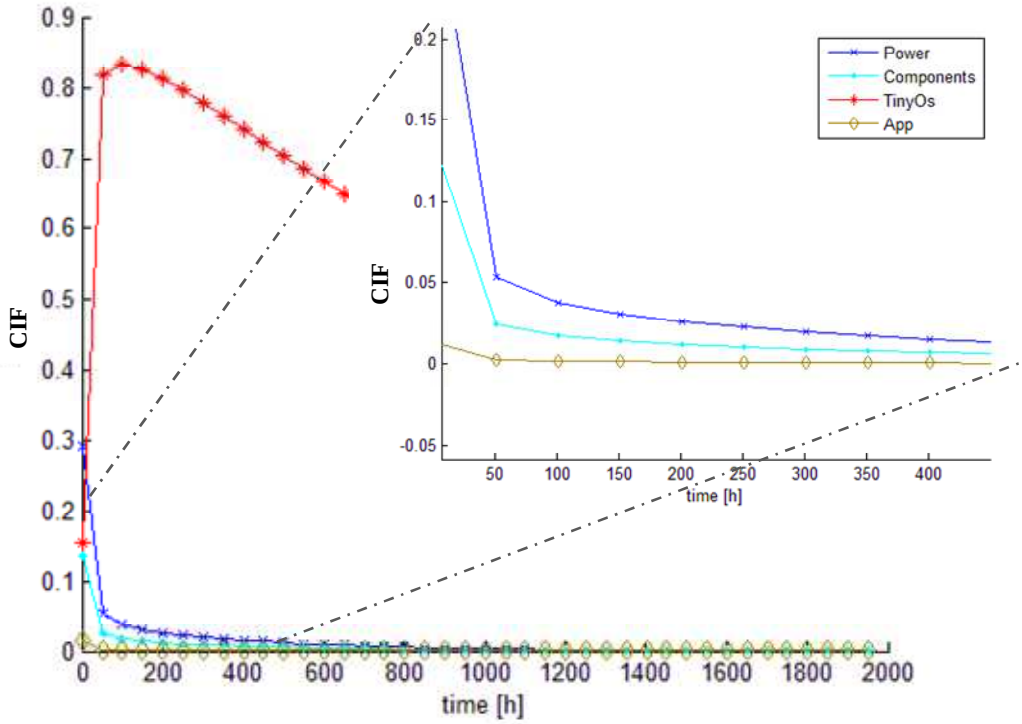


Figura 3.21: CIF dei dispositivi della Sensor Network

Interessante è la misura che riguarda la RRW (Figura 3.22). Come già detto, la RRW^{Xi} del generico componente X_i è il coefficiente moltiplicativo con cui diminuisce l'inaffidabilità globale del sistema se il componente è completamente funzionante. Nel caso dell'esempio trattato si vede che, dopo un periodo transitorio, il valore dell'inaffidabilità del sistema dipenda prevalentemente dal TinyOS e che, a regime, il suo perfetto funzionamento comporterebbe una diminuzione di inaffidabilità totale pari a circa il 40% del valore reale. Gli altri componenti non hanno lo stesso peso sul sistema.

Diversamente dalla RRW, la RAW^{Xi} (Figura 3.23) del generico componente X_i è il fattore moltiplicativo con cui aumenta l'inaffidabilità del sistema se il componente è completamente guasto. Si può osservare che non c'è differenza per le RAW dei vari componenti in quanto, data la presenza di sole porte OR, il guasto di uno solo dei componenti comporta il guasto dell'intero sistema.

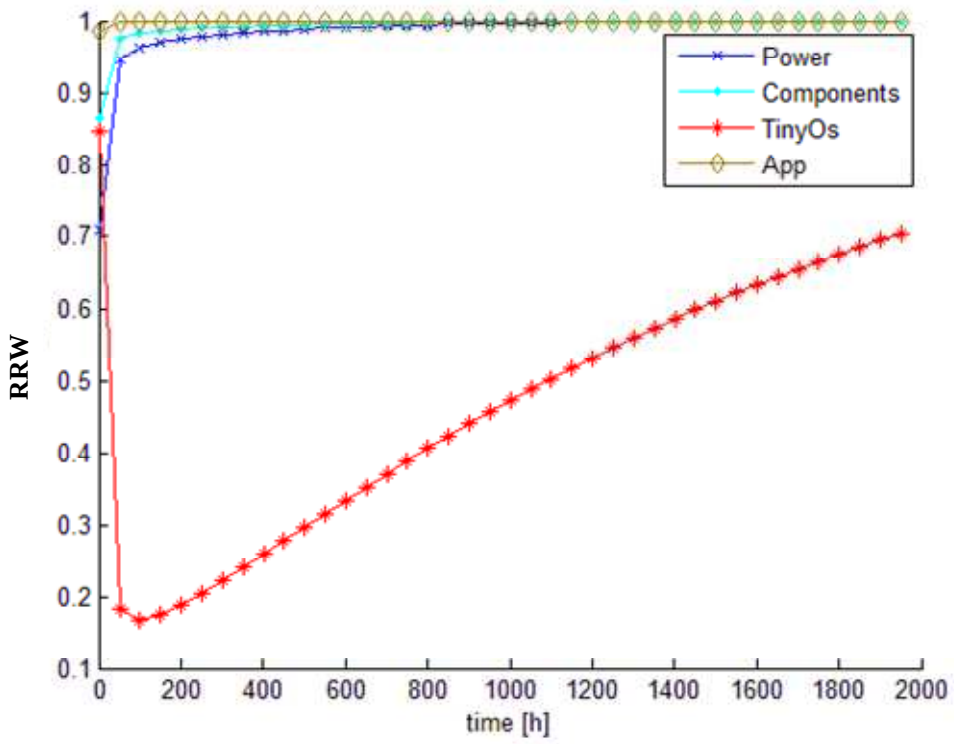


Figura 3.22: RRW dei dispositivi della Sensor Network

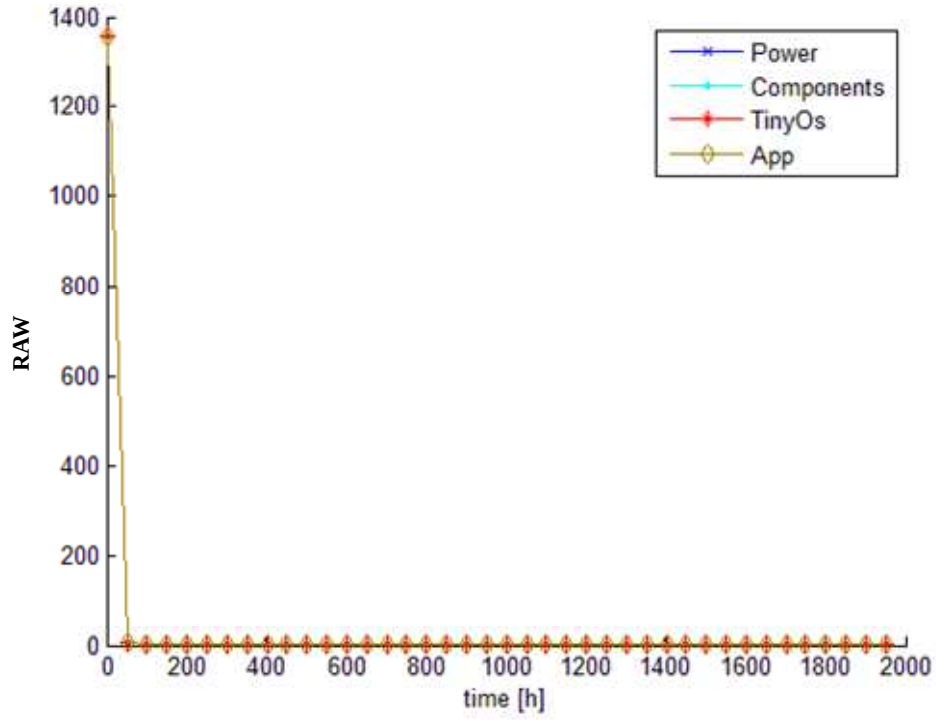


Figura 3.23: RAW dei dispositivi della Sensor Network

Le IMs così calcolate possono essere usate per effettuare l'attività di classificazione di importanza dei vari dispositivi della Sensor Network. Questa classificazione può essere diversa a seconda dell'istante t a cui si è interessati (ricordiamo che le IMs sono funzioni del tempo). Nelle Tabelle 3.11, 3.12 e 3.13 sono riportati i valori delle IMs dei componenti della Sensor Network calcolate per tempi di missione differenti, rispettivamente a 50h, 200h e 800h. Nelle stesse tabelle, i componenti vengono ordinati per ordine di importanza (dal più importante - in rosso - al meno importante - in verde): si fa osservare che, fin dai primi istanti, le IMs confermano quanto ipotizzato dalla sola osservazione del grafico delle inaffidabilità (Figura 3.19). Infatti, il TinyOS è il sistema a rilevanza maggiore per l'inaffidabilità della Sensor Network, seguito dal sistema di alimentazione (Power), dai componenti e infine dai software di Application.

Tabella 3.11: IMs al tempo di missione T_m = 50h

IM\Comp	TINYOS	POWER	SENSOR	TRANS	MICRO	MEM	APP
BIM	0.96927	0.84101	0. 83622	0. 83622	0. 83622	0. 83622	0.83247
CIF	0.81705	0.05333	0.02483	0.02483	0.02483	0.02483	0.00247
RAW	5.95425	5.95425	5.95425	5.95425	5.95425	5.95425	5.95425
RRW	0.18294	0.94666	0.97516	0.97516	0.97516	0.97516	0.99752

Tabella 3.12: IMs al tempo di missione T_m = 200h

IM\Comp	TINYOS	POWER	SENSOR	TRANS	MICRO	MEM	APP
BIM	0.88264	0.39191	0.38306	0.38306	0.38306	0.38306	0.37623
CIF	0.81208	0.02631	0.01214	0.01214	0.01214	0.01214	0.00120
RAW	1.60122	1.60122	1.60122	1.60122	1.60122	1.60122	1.60122
RRW	0.18791	0.97368	0.98597	0.98597	0.98597	0.98597	0.99879

Tabella 3.13: IMs al tempo di missione T_m = 800h

IM\Comp	TINYOS	POWER	SENSOR	TRANS	MICRO	MEM	APP
BIM	0.60694	0.03610	0.03295	0.03295	0.03295	0.03295	0.03066
CIF	0.59460	0.00582	0.00261	0.00261	0.00261	0.00261	0.00025
RAW	1.03137	1.03137	1.03137	1.03137	1.03137	1.03137	1.03137
RRW	0.40539	0.99413	0.99738	0.99738	0.99738	0.99738	0.99974

Utilizzando le regole di (Fricks & Trivedi, 2003) mediante le *(E2.17)* ed *(E2.20)* per le CTMC dei sistemi TinyOS e Power, è possibile individuare all'interno di questi sistemi complessi i meccanismi più critici per l'intero sistema della Sensor Network. In questo modo possono essere effettuate valutazioni più profonde che possono tornare utili per le politiche di ottimizzazione e di miglioramento del sistema.

La prima cosa da fare, dunque, è individuare i meccanismi interni ai componenti di Power e TinyOS che possono condizionare il funzionamento affidabilistico.

Per il Power (vedi Figura 3.23) ne vengono individuati 3:

1. il meccanismo di stand-by (riposo) della batteria (determinato in fase di progettazione e gestito mediante il parametro relativo al tasso di riposo);
2. il guasto o l'esaurimento della batteria (che può dipendere dal tipo di batteria usata);
3. l'occorrenza di un guasto permanente (guasti di natura varia esterni e su cui non si può agire).

Sulla base di questi tre eventi, ad ogni stato della CTMC viene associata una reward-rate pari a 1 se il sistema Power funziona, 0 se è guasto. Ognuno di questi stati viene univocamente individuato dalla combinazione dei 3 eventi sopra citati che possono essere in uno stato U (= up) o D (= down). Mediante l'algoritmo FT sono valutati i valori dei δ tipici per ogni stato, mentre le corrispondenti probabilità sono calcolate attraverso la risoluzione della CTMC associata.

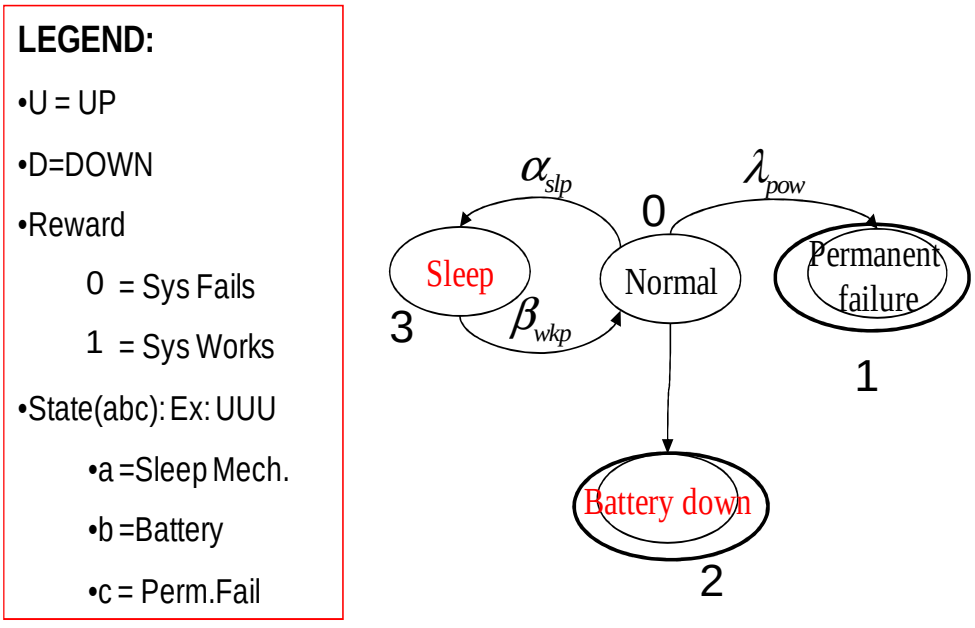


Figura 3.23: Sistema di alimentazione Power e la sua MRM associata (gli stati con il doppio cerchio sono stati di guasto, con reward-rate=1)

Tabella 3.14: parametri della tecnica FT per il calcolo della FTM del Power

ID	STATE	REWARD	δ_b	δ_{pf}	Ps(t)
0	UUU	1	1	1	$P_0(t)$
1	UDU	0	1	0	$P_1(t)$
2	UUD	0	0	1	$P_2(t)$
3	DUU	1	1	1	$P_3(t)$

Analogamente (Figura 3.24), possono essere individuati due eventi di interesse per il Sistema Operativo, TinyOS:

1. il meccanismo di identificazione degli attacchi (literal c, legenda in Figura 3.24);
2. il meccanismo di rejuvenation (vedi literal e, legenda in Figura 3.24).

I meccanismi sopra elencati risultano di indubbia importanza per l'affidabilità del TinyOS poiché la sicurezza informatica del sistema dipende dalla loro robustezza.

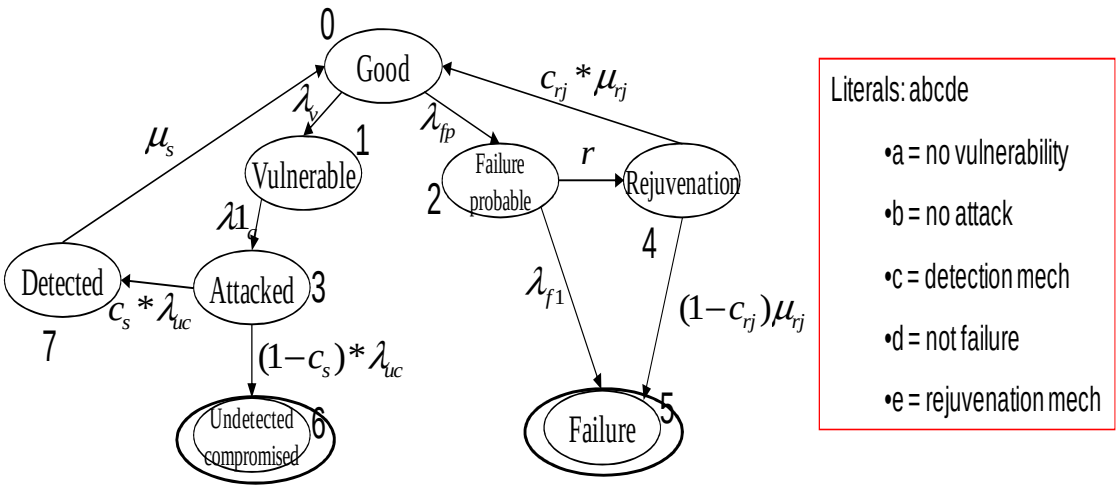


Figura 3.24: Sistema Operativo TinyOS e la sua MRM associata (gli stati con il doppio cerchio sono stati di guasto, con reward-rate=1)

In Tabella 3.15 sono riportati i parametri per il calcolo della misura FTM mediante l'algoritmo FT.

Tabella 3.15: parametri caratteristici per il calcolo della FTM del TinyOS

ID	STATE	REWARD	δ_{det}	δ_{rej}	Ps(t)
0	UUUUU	1	0	0	$P_0(t)$
1	DUUUU	1	0	0	$P_1(t)$
2	UUUUU	1	0	1	$P_2(t)$
3	DDUUU	1	1	0	$P_3(t)$
4	UUUUU	1	0	1	$P_4(t)$
5	UUUDD	0	0	1	$P_5(t)$
6	DDDUU	0	1	0	$P_6(t)$
7	DDUUU	1	1	0	$P_7(t)$

Alla luce di queste ulteriori indagini, interne ai sistemi complessi della Power System e del TinyOS, mediante la tecnica FT si può stimare il contributo dei seguenti meccanismi:

1. guasto della batteria (meccanismo interno al Power System);
2. guasto permanente del Power System (meccanismo interno al Power System);
3. fallimento del meccanismo di detection (meccanismo interno al TinyOS);
4. fallimento del meccanismo di rejuvenation (meccanismo interno al TinyOS).

In Figura 3.25 vi è il confronto tra la BIM/FTM del sistema statico e quello dello stesso sistema in cui sono considerati i contributi dei meccanismi interni ai sottosistemi composti del Power e del TinyOS: è evidente l'importanza dovuta al contributo di Detection e Rejuvenation del TinyOS e del Permanent Fault del sistema Power, rispetto all'inaffidabilità del sistema globale.

Risulta utile confrontare questi nuovi risultati con quelli calcolati precedentemente: si può osservare dalla FTM che i meccanismi interni al TinyOS diventano preponderanti soltanto dopo un transitorio di circa 200 ore quando invece, in termini di BIM, il sistema TinyOS è critico fin dall'istante iniziale del tempo di missione.

Questo comportamento richiede ulteriori approfondimenti. Infatti, osservando la CTMC del TinyOS (Figura 3.18 o Figura 3.24) si può verificare che:

1. a differenza dei Componenti e dell'Application, l'intervento dei meccanismi che portano a guasto (che nel modello del TinyOS sono appunto il fallimento della Detection e della Rejuvenation) diventano apprezzabili soltanto dopo che il sistema

transita attraverso lo stato 1 (vulnerabilità del sistema operativo) o 2 (possibile guasto). Chiaramente questa circostanza richiede del tempo, cioè il tempo che il TinyOS si trovi a fronteggiare attacchi o possibili guasti;
2. quando si calcola la BIM del modello statico, i comportamenti interni al sottosistema composto del TinyOS non sono apprezzati poiché (nel calcolo della BIM) per queste valutazioni anche la dinamica del sistema TinyOS viene semplificata rendendola analoga a quella dei Componenti (o dell'Application) mediante due soli stati (funzionante e guasto). È per questo motivo che la BIM del sottosistema TinyOS (Figura 3.20) risulta fin dall'inizio elevata.

Per il sistema di Power, invece, le BIM del "PermFault" o del "Battery" sono rilevanti all'inizio per poi perdere di importanza nel tempo. Questo comportamento si spiega guardando la CTMC del sistema Power (Figura 3.17 o Figura 3.23). Infatti per questo sottosistema, fin dall'istante iniziale sono possibili transizioni immediate verso stati di guasto e questo giustifica il motivo per cui l'importanza di questi meccanismi viene è apprezzata fin dall'inizio.
Questo stesso effetto può essere riscontrato anche per i Components che sono descritti mediante una semplice CTMC a due stati (funzionamento e guasto) per i quali esiste fin dall'istante t=0 una transizione verso lo stato di guasto.
In Figura 3.26, anche il CIF e il CFT a confronto mostrano una marcata somiglianza: la CFT consente lo studio dei meccanismi interni ai sottosistemi del TinyOS e del Power System e, anche in questo caso, si valorizza l'ipotesi che il TinyOS con il meccanismo di Detection e di Rejuvenation ha un'importanza maggiore rispetto agli altri.
Discorso analogo può essere fatto mediante il confronto tra la RRW e la FTRRW (Figura 3.26) in cui si osserva quanto il meccanismo di detection del TinyOS pesi molto più di tutti gli altri sull'affidabilità potenziale del sistema.

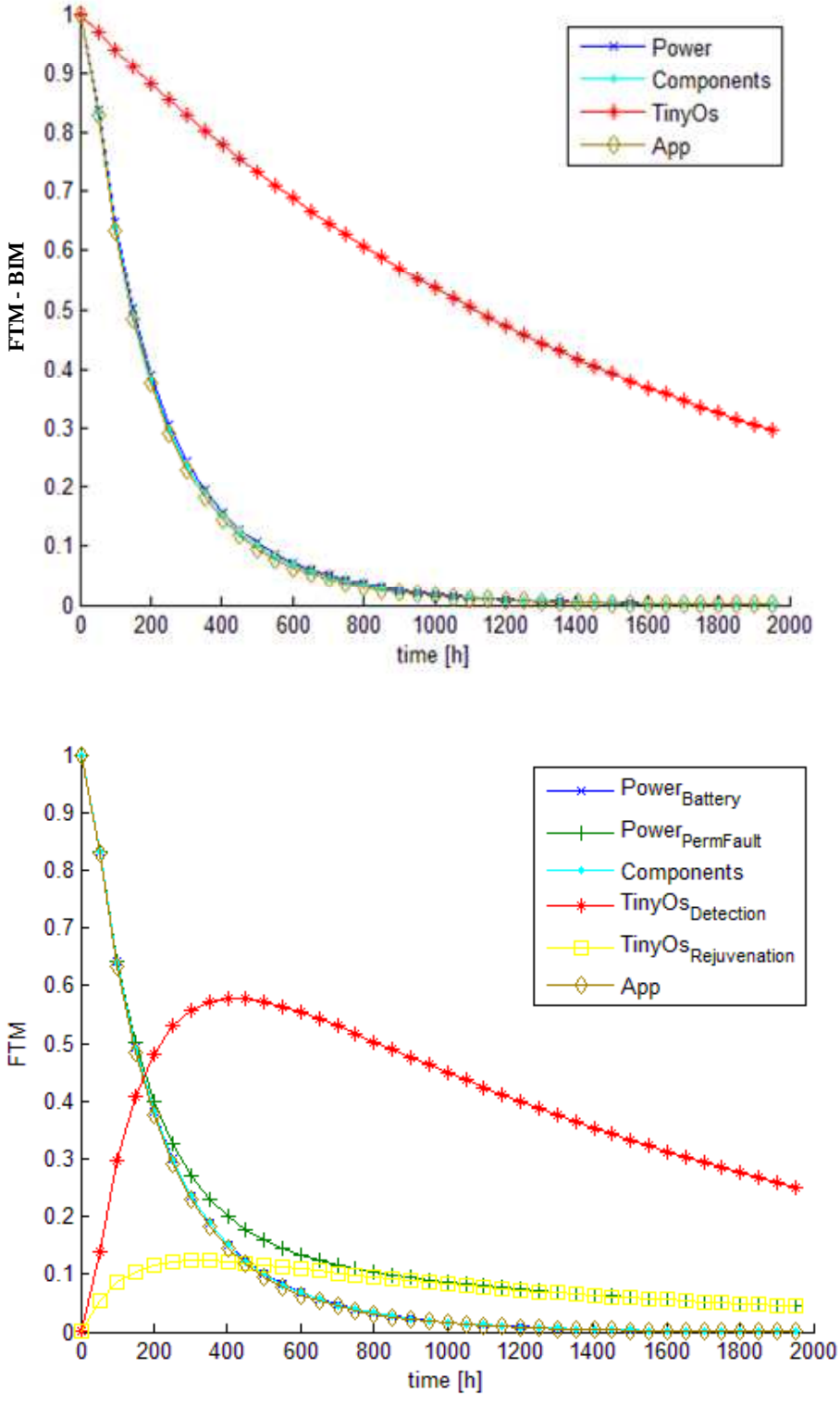


Figura 3.25: Confronto fra le FTM del sistema statico puro (in alto) e quello che tiene conto dei sottomodelli composti Power e TinyOS

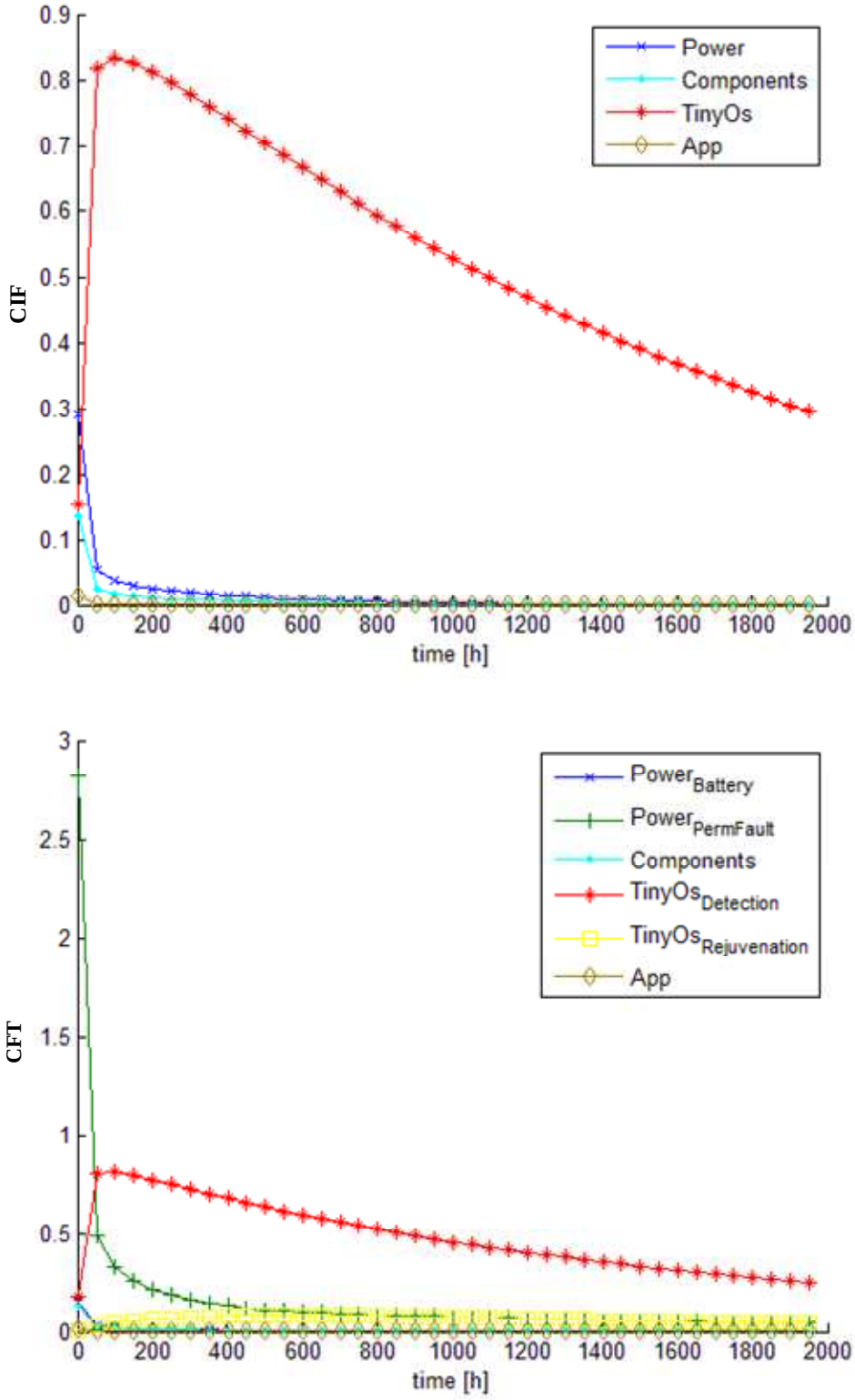


Figura 3.25: Confronto fra CIF e CFT del sistema statico puro (in alto) e quello che tiene conto dei sottomodelli composti Power e TinyOS

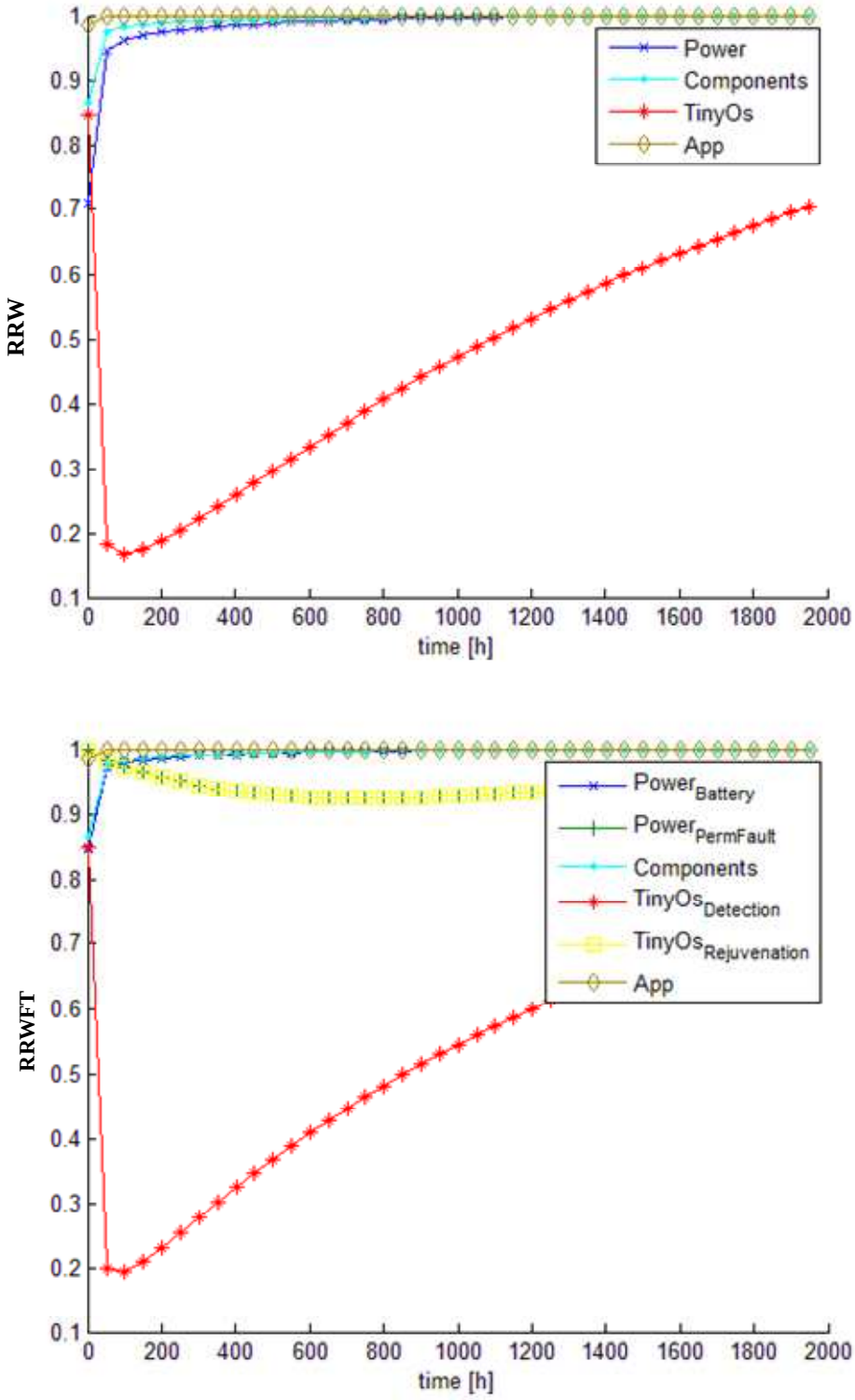


Figura 3.26: Confronto fra RRW e RRWFT del sistema statico puro (in alto) e quello che tiene conto dei sottomodelli composti Power e TinyOS

Alla luce di queste nuove valutazioni, si può effettuare una riclassificazione dei meccanismi/componenti più rilevanti per la Sensor Network, come presentato nelle Tabelle 3.16, 3.17 e 3.18.

Tabella 3.16: IMs derivate dalla FTM al tempo di missione T_m = 50h

IM\Comp	BATT	P.FAULT	SENSO	TRANS	MICR	MEM	APP	DETEC	REJUV
FTM	0. 83682	0. 83623	0. 83622	0. 83622	0. 83622	0. 83622	0.83247	0.13476	0.05279
CFT	0.02835	0.02483	0.02483	0.02483	0.02483	0.02483	0.00247	0.80243	0.01283
RAWFT	5.95425	5.95425	5.95425	5.95425	5.95425	5.95425	5.95425	1.0	1.3
RRWFT	0.97157	0.97509	0.97516	0.97516	0.97516	0.97516	0.99752	0.2	0.98268

Tabella 3.17: IMs derivate dalla FTM al tempo di missione T_m = 200h

IM\Comp	DETEC	BATT	P.FAULT	SENSOR	TRANS	MICRO	MEM	APP	REJUV
FTM	0.48089	0.38423	0.38315	0.38306	0.38306	0.38306	0.38306	0.37623	0.11576
CFT	0.77001	0.01388	0.01214	0.01214	0.01214	0.01214	0.01214	0.00120	0.0284
RAWFT	1.0	1.60122	1.60122	1.60122	1.60122	1.60122	1.60122	1.60122	1.14
RRWFT	0.230	0.98597	098771	0.98785	0.98785	0.98785	0.98785	0.99879	0.95762

Tabella 3.18: IMs derivate dalla FTM al tempo di missione T_m = 800h

IM\Comp	DETEC	REJUV	BATT	P.FAULT	SENSOR	TRANS	MICRO	MEM	APP
FTM	0.50308	0.09517	0.03447	0.033072	0.03295	0.03295	0.03295	0.03295	0.03066
CFT	0.51886	0.0477	0.00310	0.00262	0.00261	0.00261	0.00261	0.00261	0.00025
RAWFT	1.0	1.022	1.03137	1.03137	1.03137	1.03137	1.03137	1.03137	1.03137
RRWFT	0.48114	0.9224	0.98597	098771	0.99738	0.99738	0.99738	0.99738	0.99738

Come si può osservare rispetto alle valutazioni di classificazione precedente (Tabella 3.11, 3.12 e 3.13) risulta evidente che per tempi di missione bassi (T_m=[0, 200]) i meccanismi legati al TinyOS pesino meno rispetto agli altri. Il TinyOS diventa rilevante per tempi di missione più elevati. Senza un'analisi di dettaglio non

sarebbero mai potuti emergere questi importanti aspetti, legati alle dinamiche dei modelli tempo-dipendenti (come le CTMC o i DFT).

Possiamo concludere quindi che, a differenza delle tradizionali IMs per i modelli statici, il vantaggio delle misure derivate dalla FTM è di poter considerare la reale importanza di un sistema anche in funzione del tempo di missione. In questo senso, tali misure offrono degli importanti spunti per l'ottimizzazione e la riallocazione delle risorse. Nel caso della Sensor Network, per esempio, potremmo concludere che, in termini di ottimizzazione, gli investimenti sul miglioramento del sistema operativo TinyOS sarebbero giustificati soltanto se i servizi offerti dalla Sensor Network dovessero essere disponibili ininterrottamente per tempi più lunghi di 200 ore. Reti domestiche, per esempio, non devono esporre questi requisiti e, di fatto, i sistemi operativi commerciali per questo tipo di mercato non sono così affidabili.

4. CONCLUSIONI E SVILUPPI FUTURI

In questo lavoro di tesi sono stati presentati i paradigmi di risoluzione dei modelli di rischio quali i DFT e le CTMC e sono state sviluppate delle tecniche per la valutazione delle misure di importanza.
I contributi tecnici di questo lavoro di tesi possono essere sintetizzati come segue:

- l'individuazione di classi di DFT a cui associare la tecnica di risoluzione più idonea al modello di rischio;
- la definizione degli approcci di risoluzione basati sulla gerarchizzazione (esatta e non esatta);
- l'introduzione dei problemi legati alla modellazione di sistemi basati su componenti riparabili che danno luogo al calcolo della disponibilità o della prima occorrenza di un Top Event;
- l'analisi comparativa di tre importanti software per l'analisi di affidabilità (Galileo, SHARPE e Relex®) e le indicazioni per il loro utilizzo;
- l'allestimento della simulazione ad eventi discreti in ambiente Excel®, laddove risulta impossibile una risoluzione basata sui precedenti software;
- la consapevolezza della differenza dei risultati di uno stesso modello dinamico quando alimentati da componenti soggetti a guasti casuali oppure da componenti il cui tempo di guasto segue una distribuzione di Weibull;
- l'analisi delle misure di importanza per modelli DFT e modelli basati su CTMC e lo sviluppo di una tecnica basata sulla misura di Birnbaum che possa essere valida nei modelli dinamici sopra indicati.

Molti dei risultati a cui si è pervenuti non sono definitivi poiché la forma chiusa per i modelli dinamici di sistemi complessi è lungi dall'essere ricavabile. D'altra parte, le strategie studiate e le novità introdotte si inseriscono comunque all'interno di un framework le cui ipotesi risultano ancora troppo vincolanti rispetto alle caratteristiche che i sistemi reali possono descrivere.
L'insufficienza delle CTMC basate esclusivamente sulla distribuzione esponenziale negativa e della tecnica di gerarchizzazione (che sviluppa distribuzioni di probabilità

generalizzate) legittimano la ricerca di tecniche generalizzate, orientate all'utilizzo di processi semi-Markoviani.

La ricerca di metodi efficaci per la valutazione della disponibilità e della prima occorrenza di un TE, in presenza di eventi riparabili, è ancora all'inizio. Attualmente, anche grazie alla potenza dei sistemi di calcolo, la strategia più valida per questi contesti è offerta dalla simulazione: aver provato la convergenza dei risultati mediante un ambiente di simulazione *wysiwyg* (in Excel®) incoraggia lo sviluppo di software più flessibili basati sulla stessa logica ad eventi discreti.

Le problematiche relative ai modelli si ripercuotono direttamente sullo studio delle misure di importanza e della sensitività e la ricerca di risultati esatti è possibile soltanto per modelli molto semplificati. La tecnica proposta basata sul calcolo diretto della Birnbaum Measure e sull'uso delle MRM risponde all'esigenza di avere delle valutazioni più approfondite per gli studi di ottimizzazione e manutenzione nell'ambito della PRA. Si è provato che questi risultati hanno dei fondamenti tali da meritare studi più approfonditi, basati su una teoria matematica meglio definita.

I casi studio analizzati (basati su applicazioni reali) hanno messo in luce quanto l'attività di modellazione (nella scelta del modello più adeguato, del tipo di distribuzione di probabilità per il calcolo dei tempi di guasto e di indisponibilità e del livello di dettaglio) possa condizionare la risoluzione di un modello e dei suoi risultati, confermando la necessità di tecniche meglio definite e accettate.

Riallacciandomi ai problemi introdotti con la cronistoria della prefazione, concedendomi in maniera del tutto originale il rigore di un letterato, concludo affermando che la PRA è "probabilmente" uno dei pochi settori che non necessitano di rivoluzioni tecnologiche per apportare dei miglioramenti ed è tale per cui ad ogni piccolo avanzamento nella conoscenza può corrispondere un enorme beneficio per la collettività.

5. BIBLIOGRAFIA

Amari, S., Dill, G. & Howald, E., 2003. A New Approach to Solve Dynamic Fault Trees. In *PROCEEDINGS Annual RELIABILITY AND MAINTAINABILITY Symposium*., 2003. IEEE.

Anand, A. & Somani, A.K., 1998. Hierarchical analysis of fault trees with dependencies, using decomposition. In *Proceedings Annual Reliability and Maintainability Symposium*., 1998.

Assaf, T. & Dugan, J.B., 2004. Diagnostic expert systems from dynamic fault trees. In *Reliability and Maintainability, 2004 Annual Symposium*., 2004.

Aven, T. & Jensen, U., pp 8-10, 20-23, 1998. *Stochastic Models in Reliability*. Springer.

Bennets, R.G., 1975. On the analysis of fault trees. *IEEE Transactions on reliability*, pp.175-85.

Birnbaum, Z.W., 1969. *On the Importance of Different Components in a Multicomponent System*. New York: P. R. Krishnaiah, Academic Press Washington Univ Seattle Lab Of Statistical Research.

Birolini, A.,pp. 11, 25.-39, 52, 458, 483, 2003. *Reliability Engineering*. Springer.

Blake, J.T., Reibman, A.L. & Trivedi, K.S., 1988. Sensitivity analysis of reliability and performability measures for multiprocessor systems. In *SIGMETRICS '88 Proceedings of the 1988 ACM SIGMETRICS conference on Measurement and modeling of computer systems*., 1988. ACM SIGMETRICS Performance Evaluation Review.

Borgonovo, E., 2007. Differential, criticality and Birnbaum importance measure: an application to basic event, groups and SSC event in trees and binary decision diagram. *Reliability Engineering and System Safety*, pp.1458-67.

Borgonovo, E. & Apostolakis, G.E., 2001. A new importance measure for risk-informed decision making. *Reliability Engineering & System Safety*, pp.193-212.

Boudali, H., Crouzen, P. & Stoelinga, M., 2007. A Compositional Semantics for Dynamic Fault Trees in Terms of Interactive Markov Chains. In *Automated Technology for Verification and Analysis, 5th International Symposium, ATVA 2007*. Tokyo, 2007. Lecture Notes in Computer Science - Springer.

Boudali, H., Crouzen, P. & Stoelinga, M., 2007. Dynamic Fault Tree analysis using Input/Output Interactive Markov Chains. In *37th Annual IEEE/IFIP International Conference on Dependable Systems and Networks (DSN '07).*, 2007. IEEE Computer Society.

Boudali, H. & Dugan, J.B., 2005. A new Bayesian network approach to solve dynamic fault trees. In *Proocedings of Reliability and Maintainability Symposium*. Los Alamitos, 2005. IEEE.

Boyd, M.A. & Tuazon, J.O., 1991. Fault Tree Models for Fault Tolerant Hypercube Multiprocessors. In *PROCEEDINGS Annual RELIABILITY AND MAINTAINABILITY Symposium.*, 1991. IEEE.

Brown, K.S., 1990. Evaluating fault trees (AND & OR Gates only) with repeated events. *IEEE Transactions on reliability*, 39, pp.226-35.

Bryant, R.E., 1992. Symbolic boolean manipulation with ordered binary decision diagrams. *ACM Computing Surveys*, pp.293-318.

Camarinpolous, L. & Yllera, J., 1985. An improved top-down algorithm combined with modularization as a highly efficient method for fault tree analysis. *Reliability Engineering*, pp.93-102.

Chatterjee, P., 1975. Modularization of fault trees: a method to reduce the cost of analysis. In *Reliability and Fault Tree Analysis*.

Cheok, M.C., Parry, G.W. & Sherry, R.R., 1998. Use of importance measures in risk-informed regulatory applications. *Reliability Engineering and System Safety*, pp.213-26.

Compagno, L. & al., e., 2008. An on-line fault tree analysis for the continuous monitoring of the industrial plant accidents. In *Valutazione e gestione del rischio negli insediamenti civili ed industriali*. Pisa, 2008.

Coppit, D., Sullivan, K.J. & Dugan, J.B., 2000. Formal Semantics of Models for Computational Engineering: A Case Study on Dynamic Fault Trees. In *International Symposium on Software Reliability Engineering.*, 2000.

Dugan, J.B., Bavuso, S.J. & Boyd, M.A., 1992. Dynamic Fault-Tree Models for Fault-Tolerant Computer Systems. *IEEE Transactions on Reliability*, pp.363-77.

Dugan, J.B., Sullivan, K.J. & Coppit, D., 2000. Developing a Low-Cost High-Quality Software Tool for Dynamic Fault-Tree Analysis. *IEEE Transactions on Reliability*, pp.49-59.

Dugan, J.B., Venkataraman, B. & Gulati, R., 1997. DIFTree: a software package for the analysis of dynamic fault tree models. In *In Proceedings Annual Reliability and Maintainability Symposium.*, 1997.

Durga, R. & al, e., 2007. Dynamic fault tree analysis using Monte Carlo simulation in probabilistic safety assessment. *Reliability Engineering and System Safety*, pp.872-83.

Dutuit, Y. & Rauzy, A., 2000. Efficient algorithms to assess components and gates importances in Fault-Tree analysis. *Reliability Engineering and System Safety*, pp.213-22.

Ebrahimi, N., 1990. Binary Structure Functions with Dependent Components. *Advances in Applied Probability*, 22, pp.627-40.

Epstein, S. & Rauzy, A., 2004. Can We Trust PRA. *Reliability Engineering and System Safety*, pp.195-205.

Fricks, R.M. & Trivedi, K.S., 2003. Importance Analysis with Markov Chains. In IEEE, ed. *Reliability and Maintainability Symposium.*, 2003. IEEE.

Gallager, R.G., 1996. *Discrete Stochastic Processes*. Boston: Kluwer Academic Publishers.

Glynn, P.W., 1989. A GSMP Formalism for Discrete Event Systems. In *Proceedings of the IEEE.*, 1989. IEEE.

Gokhale, S.S. & Trivedi, K.S., 2002. Reliability prediction and sensitivity analysis based on software architecture. In *Software Reliability Engineering, 2002. ISSRE 2002. Proceedings. 13th International Symposium.*, 2002.

Goldfeld, A. & Dubi, A., 1987. Monte Carlo methods in reliability engineering. *Quality and Reliability Engineering International*, pp.83-91.

Gough, W.S., Riley, J. & Koren, J.M., 1990. A new approach to the analysis of the reliability block diagram. In *Proceedings Annual Reliability and Maintainability Symposium.*, 1990. IEEE.

Gulati, R. & Dugan, J.B., 1997. A modular approach for analyzing static and dynamic fault trees. In *Proceedings Annual Reliability and Maintainability Symposium.*, 1997.

IEEE, 1975. *IEEE Guide for General Principles of Reliabilitv Analysis of Nuclear Power Plant*. The Institute of Electrical and Electronics Engineers, Inc.

Khan, F.I. & Abbasi, S.A., 2000. Analytical simulation and PROFAT II: a new methodology and a computer automated tool for fault tree analysis in chemical process industries. *Journal of Hazardous Materials*, pp.1-27.

Kim, D.S., Ghosh, R. & Trivedi, K.S., 2010 in publishing. A Hierarchical Model for Reliability Analysis of Sensor Networks., 2010 in publishing.

Kohda, T., Henley, E.J. & Inoue, K., 1989. Finding modules in fault tree. *IEEE Transactions on Reliability*, pp.vol.38, 165-176.

Kosiuczenko, P. & Lajios, G., 2007. Simulation of generalised semi-Markov processes based on graph transformation systems. *Electronic Notes in Theoretical Computer Science*, pp.73-86.

Lanus, M. & Trivedi, K.S., 2003. Hierarchical composition and aggregation of state-based availability and performability models. *IEEE Transactions on Reliability*, pp.44-52.

Lee, W.S., Grosh, D.L., Tillman, F.A. & Lie, C.H., 1985. Fault Tree Analysis, Methods, and Applications - A Review. *IEEE Transactions on Reliability*, pp.194-203.

Limnios, N. & Oprisan, G., 2001. *Semi-Markov Processes and Reliability*. Birkhauser.

Limnios, N. & Ziani, R., 1986. An algorithm for reducing cut sets in fault tree analysis. *IEEE Transactions on reliability*, pp.559-563, Vol. 35, N.5.

Locks, M.O., 1978. Relationship between minimal path sets and cut sets. *IEEE Transactions on reliability*, pp.106- 107, Vol. R-27, N.2.

Malhotra, M. & Trivedi, K.S., 1995. Dependability modeling using petri-nets. *IEEE Transactions on Reliability*, pp.428-40.

Meng, F.C., 1996. Comparing the importance of system components by some structural characteristics. In *IEEE Transactions on Reliability.*, 1996. IEEE.

Merle, G., Roussel, J., Lesage, J. & Bobbio, A., 2010. Probabilistic Algebraic Analysis of Fault Trees With Priority Dynamic Gates and Repeated Events. *IEEE Transactions on Reliability*, pp.250-62.

Modarres, M., 1979. *Reliability analysis of complex technical systems using the fault tree modularization technique*. PhD Thesis - MIT, Depart. of Nuclear Eng.

Modarres, M., 2008., pp 699-717 Probabilistic Risk Assessment. In K.B. Misra, ed. *Handbook of Performability Engineering*. Springer ed. pp.ISBN:978-1-84800-130-5.

Modarres, M., Kaminskiy, M. & Krivtsov, V., pp 1-5, 13, 31, 1999. *Reliability Engineering and Risk Analysis*. New York: Marcel Dekker Inc.

Murphy, K.E. & Carter, C.M., 2003. Reliability Block Diagram Construction Techniques: Secrets to Real-Life Diagramming Woes. In *Proceedings Annual Reliability and Maintainability Symposium-Tutorial Notes*. Tampa, Florida, 2003.

Neuts, M.F., 1983. Matrix-geometric solutions in stochastic models, an algorithmic approach. *Bull. Amer. Math. Soc. (N.S.)*, 8(1), pp.97-99.

Nielses, S.F., 2009. *Continuous-time homogeneous Markov chains*. Copenhagen: UNIVERSITY OF COPENHAGEN - Department of Mathematical Sciences.

NUREG-75/014, 1975. *Reactor Safety Study: An Assessment of Accident Risks in U.S. Commercial Nuclear Power Plants*. Washington, DC 20555-0001: U.S. Nuclear Regulatory Commission.

Odeh, K. & Limnios, N., 1994. An efficient method for probability evaluation. In (Eds), J.H.&.J.P.Y. *SYSTEM MODELLING AND OPTIMIZATION*. : Springer-Verlag. p. 951–957.

Office of Safety and Mission Assurance, 2002. *Probabilistic Risk Assessment - Procedures Guide for NASA Managers and Practitioners*. Washington, DC 20546: NASA Headquarters.

Olmos, J. & Wolf, L., 1996. A modular representation and analysis of fault trees. *Nuclear Engineering and Design*, pp.531-61.

Ou, Y. & Dugan, J.B., 2000. Sensitivity Analysis of Modular Dynamics Fault Trees. In *Computer Performance and Dependability Symposium. IPDS 2000.*. Chicago, 2000. Proceedings. IEEE International.

Ou, Y. & Dugan, J.B., 2004. Modular solution of dynamic multi-phase systems. *IEEE Transactions on Reliability*, pp.499-508.

Pan, H. & Trivedi, K.S., 2001. *The Reconstruction of Sharpe*. Durham - Duke University.

Pham, H., pp 40, 544-546, 613, ed., 2003. *Handbook of Reliability Engineering*. London: Springer.

Powers, G.J. & Tompkins, F.C., 1974. Fault tree synthesis for chemical processes. *AIChE Journal*, pp.376-87.

RAC START, 2003. *The applicability of Markov analysis methods to reliability, maintainability and safety*. Rome - NY: Reliability Analysis Center, Vol.2 2003.

Rackley, L.E., 1976. *System Safety Handbook For Preparation of F-16 Fault Tree Analysis*. General Dynamics Fort Worth Division.

Rausand M., H.M., 2004. *System Reliability Theory*. Wiley Interscience.

Rauzy, A., 1993. New algorithms for fault tree analysis. *Reliability engineering and system safety*, pp.203-11.

Sahinoglu, M., Ramamoorthy, C.V., Smith, A.E. & Dengiz, B., 2004. A Reliability Block Diagramming Tool to Describe Networks. In IEEE, ed. *Reliability and Maintainability Symposium*. Los Angeles, 2004.

Sahner, R., Puliafito, A. & Trivedi, K.S., 1996. *Performance and reliability analysis of computer systems: an example-based approach using the SHARPE software package*. Kluwer Academic Publishers.

Sahner, R.A. & Trivedi, K.S., 1987. Reliability Modeling Using Sharpe. *IEEE Transactions on Reliability*, pp.186-53.

Salem, L., Apostolakis, G.E. & Okrent, D., 1978. *A computer oriented approach to fault tree construction*. UCLA-ENG-7653.

Sato, N. & Trivedi, K.S., 2007. Stochastic Modeling of Composite Web Services for Closed-Form Analysis of Their Performance and Reliability Bottlenecks. In *Fifth International Conference, September 17-20, 2007. Proceedings*. Vienna, Austria, 2007. Springer Berlin / Heidelberg.

Sharma, T.C. & Bazovsky, I., 1993. Reliability analysis of large systems by Markov techniques. In *Proceedings Annual Reliability and Maintainability Symposium.*, 1993.

Sinnamon, R.M. & Andrews, J.D., 1996. Fault tree analysis and binary decision diagrams. In *Reliability and Maintainability Symposium*. Las Vegas, 1996.

Stametalos M. (OSMA), D.H., 2002. *Probabilistic Risk Assessment Procedures Guide for NASA Managers and Practitioners*. Washington DC 20546: Office of Safety and Mission Assurance, NASA Headquarters.

Sullivan, K.J., Dugan, J.B. & Coppit, D., 1999. The Galileo fault tree analysis tool. In *In Proc. Digest of Papers Fault-Tolerant Computing Twenty-Ninth Annual International Symposium.*, 1999.

Sun, H. & Andrews, J.D., 2004. Identification of independent modules in fault trees which contain dependent basic events. *Reliability Engineering and System Safety*, pp.285-96.

Torres Toledano, J.G. & Succar, L.E.S., 1998. Bayesian Networks for Reliability Analysis of Complex Systems. In *PROGRESS IN ARTIFICIAL INTELLIGENCE — IBERAMIA 98*. Cuernevaca, Mexico, 1998. Lecture Notes in Computer Science, 1998, Volume 1484/1998, 465.

Trivedi, K.S., 2002. *Probability and statistics with reliability, queuing and computer science applications*. Wiley Interscience ed. Durham: John Wiley & Sons, Inc..

UNI, 1991. *NORMA UNI 9910 - Terminologia di affidabilità, manutenibilità e qualità di un servizio*. UNI.

van der Borst, M. & Schoonakker, H., 2001. An overview of PSA importance measures. *Reliability Engineering and System Safety*, pp.241-45.

Veeraraghavan, M. & Trivedi, K.S., 1991. An Improved Algorithm for Symbolic Reliability Analysis. *IEEE Transactions on Reliability* , pp.347-58.

Vesely, W.E., Goldberg, F.F., Roberts, N.H. & Hassl, D.F., 1981. *Fault Tree Handbook*. Washington DC: US Nuclear Regulatory Commission.

Vinod, G., Kushwaha, H.S., Verma, A.K. & Srividya, A., 2003. Importance measures in ranking piping components for risk informed in-service inspection. *Reliability Engineering and System Safety*, pp.107-13.

Vrignat, P., Avila, M., Duculty, F. & Kratz, F., 2008. Conventional approaches to the modelling of a dysfunctional process in the context of maintenance activity. In IEEE, ed. *MELECON, Electrotechnical Conference*. Ajaccio, 2008.

Wagner, D.P., Cate, C.L. & Fussel, J.B., 1978. Common cause failure analysis methodology for complex systems. In Fussel, J.B. & Burdick, G.R. *Nuclear systems*

reliability - Engineering and risk assessment. Society for Industrial & Applied Mathematics,U.S. pp.289-313.

Wang, W., Loman, J. & Vassiliou, P., 2004. Reliability Importance of Components in a Complex System. In *Reliability and Maintainability, Annual Symposium*. New York, 2004. IEEE.

Watson, H.A., 1962. *Launch Control Safety Study*. BELL Telephone Laboratories.

WG 10.4 - Dependable Computing and Fault Tolerance, 1994. *Dependability: Basic Concepts and Terminology*. IFIP.

Wilson R., C.E., 2001. *Risk/Benefit Analysis*. Boston: Harvard Press.

Wilson, J.M., 1985. Modularizing and minimizing fault trees. *IEEE Transactions on reliability*, pp.320-22.

Xing, L. & Amari, S.V., 2008. Fault Tree Analysis. In Misra, K.B. *Handbook of Performability Engineering*. Springer. pp.595-619.

Xu, H. & Dugan, J.B., 2004. Combining Dynamic Fault Trees and Event Trees for Probabilistic Risk Assessment. In *Reliability adn Maintainability Annual Symposium*., 2004. IEEE.

Younes, H.L. & Simmons, R.G., 2004. Solving Generalized Semi-Markov Processes using Continuous Phase-Type Distributions. In *In Proceedings of the Nineteenth National Conference on Artificial I.* San Jose, 2004. AAAI Press.

Zakharova, Y.A. et al., 1999. Reliability of Chemical Processes in Chemical Engineering. *Chemical and Petroleum Engineering* , 35, pp.25-27.

Zang, X., Sun, H. & Trivedi, K.S., 2002. *A BDD-Based Algorithm for Reliability Graph Analysis*. Online: Available: http://citeseer.ist.psu.edu/213187.html.

APPENDICE A

CENNI DI TEORIA MATEMATICA APPLICATA ALLA DISCIPLINA DELL'AFFIDABILITÀ

L'affidabilità e la disponibilità di un sistema giocano un ruolo fondamentale nelle fasi di quantificazione della PRA (Figura A.10) perché forniscono misure quantitative sulla qualità del sistema/processo e delle sue parti (Modarres et al., 1999). Grazie a queste valutazioni, le attività della PRA possono concentrarsi sugli elementi più critici e adottare le misure più adeguate per ottenere sistemi altamente performanti. Dato il carattere aleatorio di queste proprietà, le funzioni di affidabilità e disponibilità vengono matematicamente definite per mezzo di variabili stocastiche.
In questa sezione dell'appendice sono riportate le nozioni di teoria della probabilità che permettono la definizione delle funzioni di affidabilità, disponibilità e delle misure più importanti per le valutazioni di rischio.

TEORIA DELLA PROBABILITÀ ESSENZIALE

Definizione A.1: Una famiglia A di parti di un insieme Ω si dice σ-algebra se:
1. $\emptyset, \Omega \in$ A;
2. $se\ X \in$ A, allora $X^C \in$ A;
3. $se\ X_n \in$ A, con n $= 1, \ldots$ allora:

$(a) \bigcup_{i=1}^{\infty} X_i \in$ A;

$(b) \bigcap_{i=1}^{\infty} X_i \in$ A;

Definizione A.2: Sia Ω un insieme, A una σ-algebra di parti di Ω. Una probabilità P su A è un'applicazione $\boldsymbol{P}: \mathbf{A} \rightarrow \mathbb{R}^+$ tale che

1. P(Ω) = 1;
2. se $\{X_n\}$ è una successione di elementi di A a due a due disgiunti, allora (σ-additività)

$$\boldsymbol{P}(\bigcup_{i=1}^{\infty} \boldsymbol{X_i}) = \sum_{i=1}^{\infty} \boldsymbol{P}(\boldsymbol{X_i}) \qquad \textit{(E.A.1)}$$

Definizione A.3: Chiameremo spazio di probabilità la terna (Ω, A, P), dove:

1. Ω è un insieme, detto spazio campionario (cioè l'insieme di tutti i possibili risultati);
2. A è uno spazio degli eventi di Ω, cioè una σ-algebra di parti definita su Ω;
3. P è una regola che associa un valore di probabilità ad un evento.

Gli spazi di probabilità così definiti sono spazi misurabili, per cui le probabilità rappresentano delle misure. Inoltre, essi sono modelli di situazioni non deterministiche, per cui la modellazione del fenomeno non è univoca e non esiste uno spazio di probabilità privilegiato che lo descrive.

Definizione A.4: La definizione classica di probabilità, definisce la probabilità di un evento A come il rapporto tra il numero di casi favorevoli all'evento e il numero dei casi possibili. Quindi:

$$\mathbf{P(A)} = \frac{\textbf{casi favorevoli}}{\textbf{casi possibili}} = \frac{\mathbf{n}}{\mathbf{N}} \qquad \textit{(E.A.2)}$$

PROPRIETÀ DEGLI SPAZI DI PROBABILITÀ

Si osservi che $se\ X \in \mathrm{A}, \text{allora}\ X^C \in \mathrm{A}$ e $X \cup X^C = \mathrm{A}$. Se $Y \in \mathrm{A}, \text{allora}\ \mathrm{Y} = \mathrm{Y} \cap (\mathrm{X} \cup \mathrm{X^C}) = (\mathrm{Y} \cap \mathrm{X}) \cup (\mathrm{Y} \cap \mathrm{X^C})$ e gli eventi $(\mathrm{Y} \cap \mathrm{X})$ e $(\mathrm{Y} \cap \mathrm{X^C})$ sono disgiunti. Quindi,

$$\mathbf{P(Y)} = \mathbf{P(Y \cap X)} + \mathbf{P(Y \cap X^C)} \qquad \textit{(E.A.3)}$$

1. Se $X \in \mathrm{A}$ allora,

$$\boldsymbol{P(X^C)} = \mathbf{1} - \boldsymbol{P(A)} \qquad \textit{(E.A.4)}$$

2. Se $X \subseteq \mathrm{Y}$ allora,

$$\boldsymbol{P(X)} \leq \boldsymbol{P(Y)} \qquad \textit{(E.A.5)}$$

3. Dalla formula di De Morgan $\bigcup_n X_n = (\bigcap_n X_n^C)^C$, si ricava

$$\boldsymbol{P}(\bigcup_{\boldsymbol{n}} \boldsymbol{X_n}) = \mathbf{1} - \boldsymbol{P}(\bigcap_{\boldsymbol{n}} \boldsymbol{X_n^C}) \qquad \textit{(E.A.6)}$$

4. Probabilità della riunione di più eventi (non necessariamente disgiunti):

$$\mathbf{P(X \cup Y)} = \boldsymbol{P(X)} + \boldsymbol{P(Y \cap X^C)} \qquad \textit{(E.A.7)}$$

(poiché X e $(Y \cap X^C)$ sono disgiunti e $X \cup (Y \cap X^C) = X \cup Y$) si ha

$$\mathbf{P(X \cup Y) = P(X) + P(Y) - P(X \cap Y)} \qquad \textit{(E.A.8)}$$

In generale, per la riunione di un numero finito qualunque di eventi, usando ripetutamente la *(E.A.8)* si ha:

$$\boldsymbol{P(\cup_i A_i) = \sum_{1\le i\le m} P(A_i) - \sum_{1\le i<j\le m} P(A_i \cap A_j) + \sum_{1\le i<j<k\le m} P(A_i \cap A_j \cap A_k) + \cdots + (-1)^{m-1} P(A_1 \cap A_2 \cap \ldots \cap A_k)} \qquad \textit{(E.A.9)}$$

PROBABILITÀ CONDIZIONALE

Sia (Ω, A, P) uno spazio di probabilità.

Definizione A.5: Dati $\boldsymbol{X, Y \in A}$ $\boldsymbol{e}$ $\boldsymbol{P(X) > 0}$, si chiama probabilità condizionale di $\boldsymbol{Y}$ rispetto ad $\boldsymbol{X}$ la quantità

$$\boldsymbol{P(Y|X) = \frac{P(X \cap Y)}{P(X)}} \qquad \textit{(E.A.10)}$$

Intuitivamente, si tratta della probabilità che un evento Y si verifichi data la condizione che l'evento X sia sempre verificato.

TEOREMA DI BAYES

Sia (Ω, A, P) uno spazio di probabilità.

Teorema A.1: Siano X_1, X_2, …, X_n eventi disgiunti e tali che $\mathbf{X_1 \cup X_2 \cup \ldots \cup X_n = \Omega}$. Vale la formula per un certo evento Y condizionante:

$$\mathbf{P(X_i|Y) = \frac{P(X_i)P(Y|X_i)}{P(Y)} = \frac{P(X_i)P(Y|X_i)}{\sum_{k=1}^{n} P(X_k)P(Y|X_k)}} \qquad \textit{(E.A.11)}$$

EVENTI INDIPENDENTI

Definizione A.6: Due eventi X e Y si dicono indipendenti se

$$\boldsymbol{P(X \cap Y) = P(X) \cdot P(Y)} \qquad \textit{(E.A.12)}$$

Dalla *(E.A.10)* si deduce che se due eventi X e Y sono indipendenti, la *(E.A.12)* diventa $P(Y|X) = P(Y)$.

Definizione 5.7: Si dice che X_1, X_2, …, X_n sono a due a due indipendenti se e solo se,

$$\boldsymbol{P(X_i \cap X_j) = P(X_i) \cdot P(X_j), \forall\, i, j = 1, 2, \dots, n\,, i \neq j} \qquad \textit{(E.A.13)}$$

La proprietà di indipendenza è spesso usata per definire e contestualizzare dei modelli di probabilità; infatti la maggior parte dei processi stocastici è caratterizzata da una qualche forma di indipendenza o indipendenza condizionata (Gallager, 1996).

VARIABILI ALEATORIE

I risultati di una collezione di prove sperimentali, per esempio la misura di un valore di tensione, di temperatura, ecc., non è deterministico e generalmente varia a seconda delle condizioni in cui viene eseguita la prova sperimentale. Esiste, quindi, una corrispondenza tra gli elementi dello spazio campione (i possibili risultati dell'esperimento) e lo spazio dei numeri reali. Più precisamente, una variabile aleatoria X (o stocastica, o casuale V.C.) è definita come una funzione che mappa ogni punto dello spazio campione Ω in un set di valori reali, attraverso una misura di probabilità. Dunque, per ogni variabile aleatoria X e ogni numero reale x, esiste un evento X≤x; questo evento non è altro che il sottoinsieme di Ω i cui elementi sono mappati tramite X in valori minori o al massimo uguali a x, cioè:

$$P(\mathrm{X} \leq \mathrm{x}) = P(\{\omega \in \Omega : X(\omega) \leq x\}) \qquad \textit{(E.A.14)}$$

Si osservi che P(X≤x) è una funzione della variabile reale x, monotona non decrescente a valori compresi tra [0, 1], con x che può variare da] -∞, +∞ [.

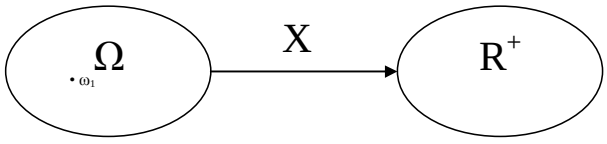


Figura A.1: Rappresentazione di una variabile aleatoria

Tale misura di probabilità viene chiamata *funzione di distribuzione di probabilità* $F_X(x)$ della V.C. (o funzione di distribuzione cumulata (CDF) o funzione di ripartizione).

Per esempio, si considerino gli esperimento del lancio di un dado e della scelta casuale di un giorno da lunedi a sabato. In questi due casi si ha $\Omega_1 = \{1, 2, 3, 4, 5, 6\}$ e $\Omega_2 = \{\text{Lun, Mar, Mer, Gio, Ven, Sab}\}$. Dal punto di vista della modellazione, i due esperimenti sono identici perché ad ogni evento di Ω_1 e Ω_2 si può associare un valore numerico discreto da 1 a 6. In questo modo l'esito dell'estrazione si valuta rispetto ai possibili risultati in $\mathbb{R}$ (in questo caso nel sottoinsieme dei numeri naturali compresi tra 1 e 6). Alla V.C. viene associata inoltre una legge di probabilità. Nota la legge di probabilità e la V.C., si può costruire la funzione di distribuzione $F_X(x)$ della V.C. (come in Figura A.2).

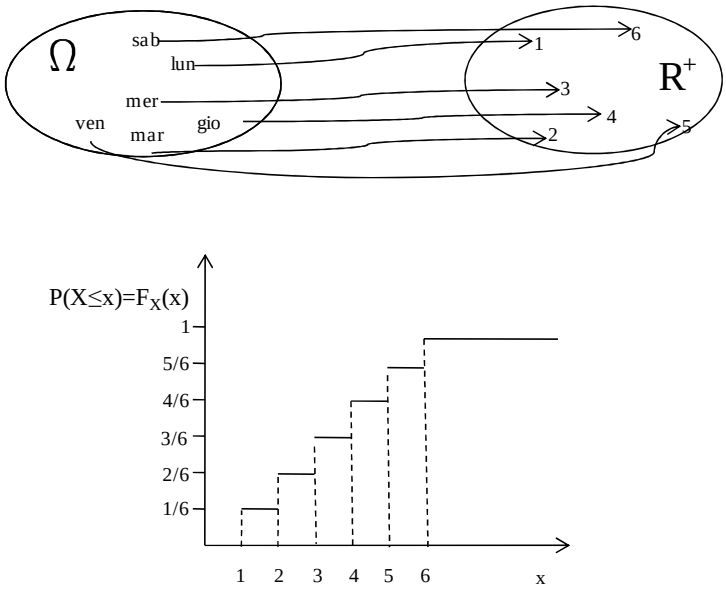


Figura A.2: Costruzione della variabile aleatoria e della funzione di ripartizione discreta per l'estrazione equiprobabile di un giorni della settimana (domenica esclusa)

Se la funzione di distribuzione $F_X(x)$ di una V.C. è derivabile, la sua derivata $f_X(x)$ è chiamata *densità di probabilità (PDF)* della V.C.. Per valori sufficientemente piccoli δ, il prodotto $\delta f_X(x)$ approssima la probabilità che X sia mappata nell'intervallo $[x, x+\delta]$. Se la densità esiste ed è finita ovunque, la V.C. è continua e si ha:

$$P(X \leq x) = F_X(x) = \int_{-\infty}^{x} f_X(\xi) d\xi \qquad \textit{(E.A.1)}$$

e

$$\boldsymbol{F'_X(x) = f_X(x)} \qquad \textit{(E.A.2)}$$

Analogamente, se X ha un numero finito di possibili immagini nell'insieme dei numeri reali x_1, x_2, ecc., la probabilità di ognuno di questi elementi x_i viene indicata mediante la *funzione di massa di probabilità (PMF)* o densità discreta della V.C. $\{P_X(x_i); i \geq 1\}$. Si può scrivere:

$$\boldsymbol{f_X(x) = P(X = x) = P(\{\omega \in \Omega: X(\omega) = x\})} \qquad \textit{(E.A.3)}$$

Diversamente dalla PDF, la PMF è un valore di probabilità (la probabilità che la variabile casuale X sia uguale ad x). Riassumendo, una distribuzione di probabilità è un modello matematico che collega il valore di una variabile alla probabilità che tale valore si riscontri all'interno dello spazio campione.
Vi sono due tipi di distribuzioni di probabilità:

1. ***distribuzioni discrete***: la variabile può assumere solo valori discreti;
2. ***distribuzione continua***: la variabile può assumere valori continui.

Definizione A.8: Le variabili casuali X_1, X_2,…, X_n sono dette indipendenti se per tutti gli x_1, x_2,…, x_n si ha:

$$\boldsymbol{F(x_1, x_2 \dots, x_n) = \prod_{i=1}^{n} P(X_i \leq x_i)}$$

(E.A.18)

Nell'ambito affidabilistico, una distribuzione congiunta modella la composizione degli scenari che conducono all'occorrenza di un evento di particolare rilevanza, di cui ha senso calcolare la probabilità. La *(E.2.18)*, dunque, è molto importante poiché consente la semplificazione del calcolo della probabilità della distribuzione congiunta di più eventi.

FUNZIONI DI AFFIDABILITÀ

Gli studi di affidabilità si basano su valutazioni di tipo probabilistico; i valori che si attribuiscono a parametri quali il tempo di guasto (T_f), l'affidabilità (R),

l'inaffidabilità (F), la disponibilità (A), l'indisponibilità (Q), il MTTF o il MTTR di un componente, derivano normalmente dall'elaborazione statistica dei risultati delle "prove di vita" eseguite su un campione statistico, cioè su un insieme di componenti estratto da una popolazione più ampia.

TEMPO DI GUASTO

Il tempo di guasto T_f di un componente è la durata di tempo complessiva del tempo di funzionamento di un'entità, dal momento in cui essa viene messo in servizio (che per definizione viene fissato al tempo t=0) o dal momento in cui viene messa disponibile, fino alla riapparizione del guasto successivo (UNI, 1991). Dato il carattere aleatorio del tempo di guasto, è intuitivo associare al tempo di guasto una V.C. che può assumere valori compresi in [0, t]. La relazione tra la variabile di stato X(t) e il tempo di guasto T_f è illustrata in Figura 2.3 (Rausand M., 2004).

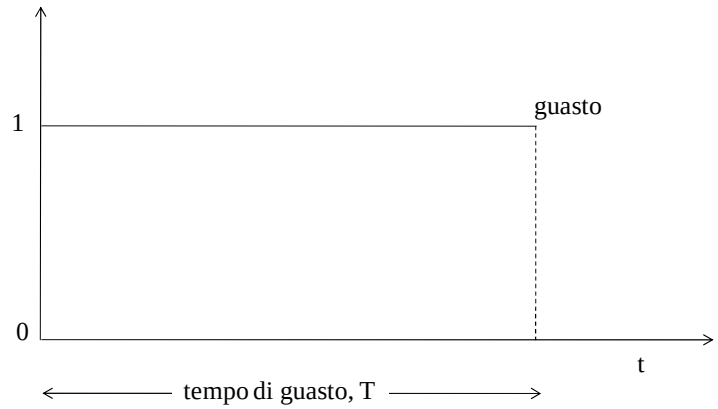


Figura A.3: relazione tra tempo di guasto e stato del componente

AFFIDABILITÀ (R) E INAFFIDABILITÀ (F)

L'affidabilità di un'entità è l'attitudine dell'entità a svolgere una funzione richiesta in condizione date per un dato intervallo di tempo (UNI, 1991):

1. si assume generalmente che l'entità sia in uno stato nel quale svolga questa funzione richiesta all'inizio dell'intervallo di tempo;
2. il termine dell'affidabilità è anche usato come misura di questa attitudine.

L'affidabilità di un dato dispositivo (oggetto, sistema o componente) viene valutata attraverso una misura di probabilità (è un valore compreso tra 0 e 1): essa rappresenta la probabilità che un sistema fornisca senza interruzioni le prestazioni richieste durante l'intervallo di tempo assegnato, fissate le condizioni operative ed ambientali di funzionamento.

Per definizione, affidabilità ed inaffidabilità sono l'una il complemento ad uno dell'altra. Per studiare l'affidabilità di un componente vengono fissate delle condizioni al contorno:

- un intervallo di tempo (tempo di missione);
- le condizioni operative alle quali è sottoposto;
- le condizioni ambientali.

Il carattere aleatorio dell'affidabilità emerge per il fatto di valutare il comportamento del dispositivo in condizioni più generali rispetto a quelle operative ed ambientali sopra citate; infatti, durante il normale funzionamento di un sistema possono sopraggiungere tutta una serie di condizioni anomale (i gusti o i disturbi esterni) la cui comparsa è del tutto casuale.

La definizione di affidabilità presuppone:

1. che sia fissato in modo univoco il criterio (C) per giudicare se l'elemento è funzionante o non funzionante. In questo lavoro di tesi, saranno investigati maggiormente i sistemi bistabili, a due soli stati di funzionamento possibili: guasto o funzionante. Per altri sistemi è possibile individuare anche stati di funzionamento parziale che rappresentano vari livelli di prestazione o degradazione (Boyd & Tuazon, 1991); in questi casi lo stato di guasto è definibile una volta che venga fissato un limite ammissibile al di sotto del quale si parla di guasto (es.: l' intensità di una sorgente luminosa);
2. che le condizioni ambientali (A) d'impiego siano stabilite e mantenute costanti nel periodo di tempo in questione;
3. che sia definito l'intervallo di tempo T_m (tempo di missione) durante il quale si richiede che il componente funzioni.

Definizione A.9: Si indichi con R l'affidabilità (Reliability). Per quanto anticipato, R è funzione di tre variabili. Dunque possiamo scrivere: R = R(C, A, t). Nel prosieguo,

fissati C ed A considereremo l'affidabilità una funzione del solo tempo, cioè R = R(t).

Per analogia, essendo l'inaffidabilità F (Failure) il complemento ad uno di R, si ha F(C,A,t) = F(t) = 1-R(t).

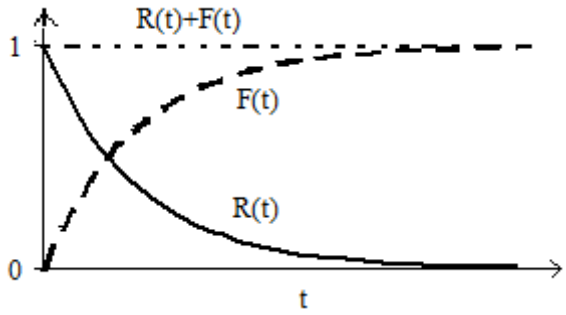


Figura A.4: Relazione tra una generica funzione di affidabilità e di inaffidabilità

Si consideri un campione di componenti elementari costituito da un grande numero N_0 di elementi uguali tutti funzionanti all'istante t = 0; indichiamo con:

- $N_v(t)$ il numero di componenti funzionanti all'istante t;
- $N_g(t)$ il numero di componenti guasti all'istante t;

La seguente uguglianza è sempre verificata: $N_0 = N_v(t) + N_g(t)$.

Utilizzando la Definizione A.4, si ha:

$$\boldsymbol{R(t) = \frac{N_v(t)}{N_0}} \qquad \textit{(E.A.19)}$$

e

$$\boldsymbol{F(t) = \frac{N_g(t)}{N_0}} \qquad \textit{(E.A.20)}$$

L'istante t = 0 non coincide, in generale, con l'inizio dell'utilizzazione del sistema/componente, bensì rappresenta l'inizio di un periodo di impiego arbitrario.

DENSITÀ DI PROBABILITÀ DI GUASTO

Definizione A.10: Sia F(t) la funzione di inaffidabilità di un componente/sistema. Definiamo la funzione densità di probabilità di guasto f(t) come:

$$\boldsymbol{f(t) = F'(t) = \frac{dF(t)}{dt}} \qquad \textit{(E.A.21)}$$

Tale funzione non è dimensionalmente una probabilità ma ha le dimensioni di $[t^{-1}]$, dal momento che è data dal rapporto tra il differenziale della funzione F (che rappresenta una probabilità infinitesima di guasto relativa all'intervallo [t, t+dt]) e l'intervallo di tempo infinitesimo dt (Figura A.5).

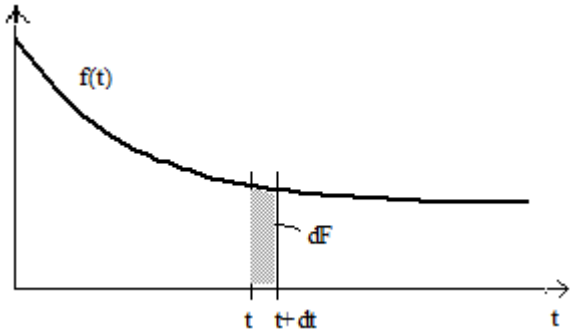


Figura A.5: Funzione di densità di probabilità di guasto

Con facili sostituzioni si ha:

$$f(t) = = \frac{dF(t)}{dt} = \frac{d}{dt}\left(\frac{N_g(t)}{N_0}\right) = \frac{1}{N_0}\left(\frac{dN_g(t)}{dt}\right) = \frac{1}{N_0}\frac{d(N_0 - N_v(t))}{dt} =$$

$$= \boldsymbol{-\frac{1}{N_0}\frac{dN_v(t)}{dt} = -\frac{dR(t)}{dt}} \qquad \textit{(E.A.22)}$$

Quindi,

$$f(t) = \frac{dF(t)}{dt} = -\frac{dR(t)}{dt} \qquad \textit{(E.A.23)}$$

È facile provare che l'istante t divide l'area sottesa dalla densità di probabilità di guasto f(t) in due zone complementari le cui aree misurano rispettivamente l'inaffidabilità F(t) e l'affidabilità R(t) del componente/sistema (vedi Figura A.6).

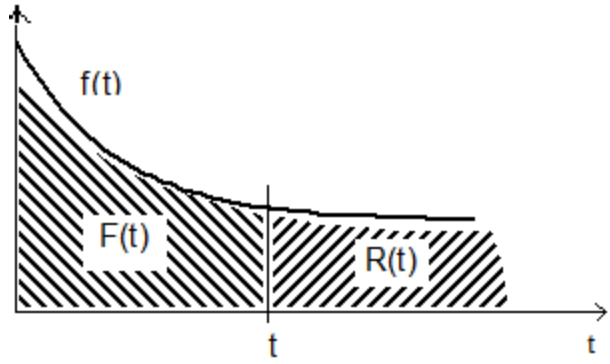


Figura A.6: Relazione tra F(t) ed R(t) rispetto alla densità di probabilità di guasto f(t)

Si considerino, a tale scopo, le seguenti ipotesi al contorno:

$$\begin{cases} \boldsymbol{R(0) = 1} \\ \boldsymbol{F(0) = 0} \end{cases} \quad \boldsymbol{e} \quad \begin{cases} \boldsymbol{R(t \to \infty) = 0} \\ \boldsymbol{F(t \to \infty) = 1} \end{cases} \qquad \textit{(E.A.24)}$$

Al tempo t = 0 il componente viene qualificato completamente funzionante e la sua affidabilità tende a zero al crescere del tempo. Se si integra la *(E.A.22)*, nei due intervalli [0, t] e [t, ∞) si ottiene:

$$\int_0^t \boldsymbol{f(t)dt} = \boldsymbol{F(t)}\Big|_0^t = \boldsymbol{F(t) - F(0) = F(t)} \qquad \textit{(E.A.25)}$$

$$\int_t^\infty \boldsymbol{f(t)dt} = \boldsymbol{F(t)}\Big|_t^\infty = \boldsymbol{F(t \to \infty) - F(t) = 1 - F(t) = R(t)} \qquad \textit{(E.A.26)}$$

Quindi l'integrale esteso all'infinito della funzione densità di probabilità di guasto f(t) vale uno e la funzione f(t) è "normale".

Si noti che fino a questo punto non è stata eseguita alcuna ipotesi sulla natura delle funzioni f(t), F(t) e R(t).

FUNZIONE TASSO DI GUASTO O DI RISCHIO (ISTANTANEO)

Si consideri la *(E.A.22)*. In particolare la forma $f(t) = \frac{1}{N_0}\left(\frac{d\,N_g(t)}{dt}\right)$ è un tasso di guasto normalizzato.

Definizione A.11: Sostituendo alla precedente il valore N_0 con $N_v(t)$, indicante il numero dei componenti ancora attivi, si definisce un'altra grandezza fondamentale della teoria dell'affidabilità, il tasso di guasto (istantaneo):

$$\boldsymbol{h(t) = \frac{1}{N_v(t)}\left(\frac{d\,N_g(t)}{dt}\right)} \qquad (E.A.27)$$

La funzione h(t) rappresenta la frazione di popolazione che si guasta in un intervallo infinitesimo dt rapportata al numero dei componenti ancora funzionanti all'istante t.
Valgono le seguenti relazioni fondamentali:

$$h(t)dt = -\frac{dR(t)}{R(t)} \qquad (E.A.28)$$

$$\boldsymbol{R(t) = e^{-\int_0^t h(t)dt}} \qquad (E.A.29)$$

Per provare la *E.A.28* si consideri la *E.A.19* e si differenzi secondo le seguenti successive uguaglianze:

$$\boldsymbol{dR(t) = \frac{dN_v(t)}{N_0} = \frac{d(N_0 - N_g(t))}{N_0} = -\frac{dN_g(t)}{N_0}} \qquad (E.A.30)$$

Moltiplicando il primo e l'ultimo termine della precedente per N_0 si può riscrivere la E.*A.30*

$$\boldsymbol{-dN_g(t) = N_0 \cdot dR(t)} \qquad (E.A.31)$$

Sostituendo la *E.A.31* alla *E.A.27* si ha:

$$\boldsymbol{h(t) = -\frac{1}{N_v(t)}\left(\frac{N_0 \cdot dR(t)}{dt}\right)} \qquad (E.A.32)$$

Separando le variabili e ricordando la *E.A.19*, si ottiene la *E.A.28*.
Infine, integrando tra 0 e t con la condizione R(0) = 1, si ottiene:

$$\boldsymbol{\int_0^t h(t)dt = -[ln\,R(t) - ln\,R(0)] = -ln\,R(t)} \qquad (E.A.33)$$

Passando agli esponenziali si ricava la *E.A.29.*
Valgono le ulteriori seguenti relazioni:

$$R(t) = \frac{f(t)}{h(t)} \qquad \textit{(E.A.34)}$$

$$h(t) = \frac{R'(t)}{R(t)} \qquad \textit{(E.A.35)}$$

Nel caso particolare in cui il tasso di guasto h(t) si mantiene costante nel tempo (guasti casuali) lo indicheremo con λ.

INTERPRETAZIONE PROBABILISTICA DEL TASSO DI GUASTO

Utilizzando la definizione di probabilità condizionata, è possibile fornire un'interpretazione probabilistica del tasso di guasto.
Si calcoli la probabilità che il componente si guasti nell'intervallo di tempo [t, t+dt], condizionata alla sua sopravvivenza fino a t:

$P\{guasto\ in\ [t, t+dt] | non\ guasto\ in\ [0,t]\}$ =

$\frac{P\{guasto\ in\ [t,t+dt]\} \cap P\{non\ guasto\ in\ [0,t]\}}{P\{non\ guasto\ in\ [0,t]\}}$ = $\frac{P\{guasto\ in\ [t,t+dt]\} \cap P\{guasto\ in\ [t,\infty)\ \}}{P\{non\ guasto\ in\ [0,t]\}}$ = (vedi nota[2]) = $\frac{P\{guasto\ in\ [t,t+dt]\}}{P\{non\ guasto\ in\ [0,t]\}} = \frac{F(t+dt) - F(t)}{R(t)} = \frac{1 - R(t+dt) - 1 + R(t)}{R(t)} = -\frac{R(t+dt) - R(t)}{R(t)} = -\frac{dR(t)}{R(t)} = h(t)dt$.

Quindi, mentre f(t)dt ha il significato di probabilità infinitesima (indipendente) di guasto in dt, h(t)dt assume il significato e le dimensioni di probabilità infinitesima che il componente si guasti nell'intervallo di tempo [t, t+dt] condizionata alla sua sopravvivenza fino a t.

PARAMETRI DI AFFIDABILITÀ

MTTF (MEAN TIME TO FAILURE)

Definizione A.12: Il MTTF è il tempo medio di guasto di un componente. Corrisponde al valore atteso del tempo al guasto (UNI, 1991), dunque, si ottiene come media continua, pesata sulla probabilità, che il componente ha di guastarsi:

[2] L'evento "guasto in [t, t+dt]" è incluso (simbolo ⊂) nell'evento "guasto in [t, ∞]. Per la teoria degli insiemi se A ⊂ B si ha A ∩ B = A.

$$MTTF = \int_0^\infty t \cdot f(t)dt \qquad (E.A.36)$$

Integrando per parti si ha:

$$MTTF = \int_0^\infty t \cdot f(t)dt =$$

$$= \int_0^\infty t\left(-\frac{dR(t)}{dt}\right)dt = -\int_0^\infty t \cdot dR(t) = -\ [t \cdot R(t)]_0^{\to\infty} + \int_0^\infty R(t)dt$$

Essendo il $\lim_{t\to\infty} t \cdot R(t) = 0$, si ha:

$$MTTF = \int_0^\infty R(t)\ dt \qquad (E.A.37)$$

cioè il tempo medio fino al guasto è pari all'area sottesa alla curva della funzione affidabilità.

CASO PARTICOLARE

Se il tasso di guasto è costante h(t) = λ (guasti casuali) si ha:

$$MTTF = \int_0^\infty e^{-\lambda t}dt = -\left[\frac{e^{-\lambda t}}{\lambda}\right]_0^\infty = \frac{1}{\lambda} \qquad (E.A.38)$$

Inoltre, sostituendo alla variabile t della $\boldsymbol{R(t) = e^{-\lambda t}}$, il valore del MTTF della *E2.38* si ha R(MTTF) = e^{-1} = 0,3679, cioè la probabilità di superare senza guasto un intervallo di tempo pari al MTTF è pari solo al 37% circa.

DISPONIBILITÀ

La disponibilità A(t), è la probabilità che un'entità sia in grado di eseguire una funzione richiesta nelle condizioni assegnate e ad un dato istante, assumendo che vengano messi a disposizione i mezzi esterni necessari .

Questa funzione si definisce nel caso in cui i sistemi/componenti siano riparabili. Rispetto all'affidabilità, in cui gli interventi di manutenzione devono essere eseguiti in intervalli di tempo non coincidenti con i tempi di missione, per sistemi riparabili la

manutenzione rende il sistema non disponibile anche per tutto il tempo necessario alla sua riparazione. La disponibilità è quindi una funzione che tiene conto sia dell'affidabilità del sistema sia degli aspetti manutentivi.
I problemi di affidabilità possono allora essere trattati come casi particolari di quelli di disponibilità, per i quali il passaggio allo stato di guasto non consente il ritorno allo stato di funzionamento (Aven & Jensen, 1998).

MTTR (MEAN TIME TO REPAIR)

Nel caso di componenti riparabili diventa fondamentale il parametro che esprime il tempo medio che intercorre tra l'insorgenza di un guasto e la sua riparazione; esso viene detto appunto "Mean Time To Repair" e si indica con la sigla MTTR. Esso corrisponde con il valore atteso del tempo al ripristino (UNI, 1991).
Per poterlo definire in analogia al MTTF possiamo fare riferimento a funzioni che sono le analoghe di quelle già definite per l'affidabilità (vedi Tabella A.1).

Tabella A.1: Analogia tra le funzioni di manutenibilità e le funzioni di affidabilità

Funzioni di manutenibilità		**Analoghe funzioni affidabilistiche**	
g(t)	densità di probabilità di riparazione (normale)	f(t)	distribuzione di probabilità di guasto
M(t)	probabilità di riparazione (manutenibilità)	F(t)	Inaffidabilità
N(t)	probabilità di non riparazione	R(t)	Affidabilità
z(t)	tasso di riparazione (istantaneo)	h(t)	tasso di guasto istantaneo

Per tali funzioni valgono relazioni formalmente identiche a quelle viste nell'ambito puramente affidabilistico; perciò, indicando con t = 0 l'istante al quale si è verificato il guasto, si ha:

- $g(t)dt = probabilità\ che\ la\ riparazione\ termini\ nell'intervallo\ [t, t+dt]$;
- $\int_0^\infty g(t)dt = 1$;
- $M(t) = probabilità\ che\ la\ riparazione\ termini\ nell'intervallo\ [0,t] = \int_0^t g(t)dt$;
- $g(t) = \frac{dM(t)}{dt} = -\frac{dN(t)}{dt}$;

- $MTTR = \int_0^\infty t \cdot g(t)dt =$

$$= \int_0^\infty t \cdot \left(-\frac{dN(t)}{dt}\right) dt = -\int_0^\infty t \cdot dN(t) =$$

$$= -[t \cdot N(t)]_0^{\to\infty} + \int_0^\infty N(t)dt = \int_0^\infty N(t)dt \qquad \textit{(E.A.39)}$$

Per attribuire un significato corretto a questo parametro si può dire che esso rappresenta l'equivalente del MTTF per componenti riparabili.

Osservando la Figura A.7, si evidenzia che la notazione utilizza due nomi diversi per indicare parametri che sono concettualmente identici: MTTF(sist. non riparabile) ≡ MTBF (sist. riparabile). Ciò che è stato indicato con MTTR talvolta viene indicato in letteratura come MDT (Mean Down Time) termine che comprende oltre al downtime per la manutenzione vera e propria (MTTR) anche ritardi nel ripristino del funzionamento, dovuti a questioni logistiche e/o amministrative (per es. chiamata della squadra di manutenzione, reperimento di componenti di ricambio).

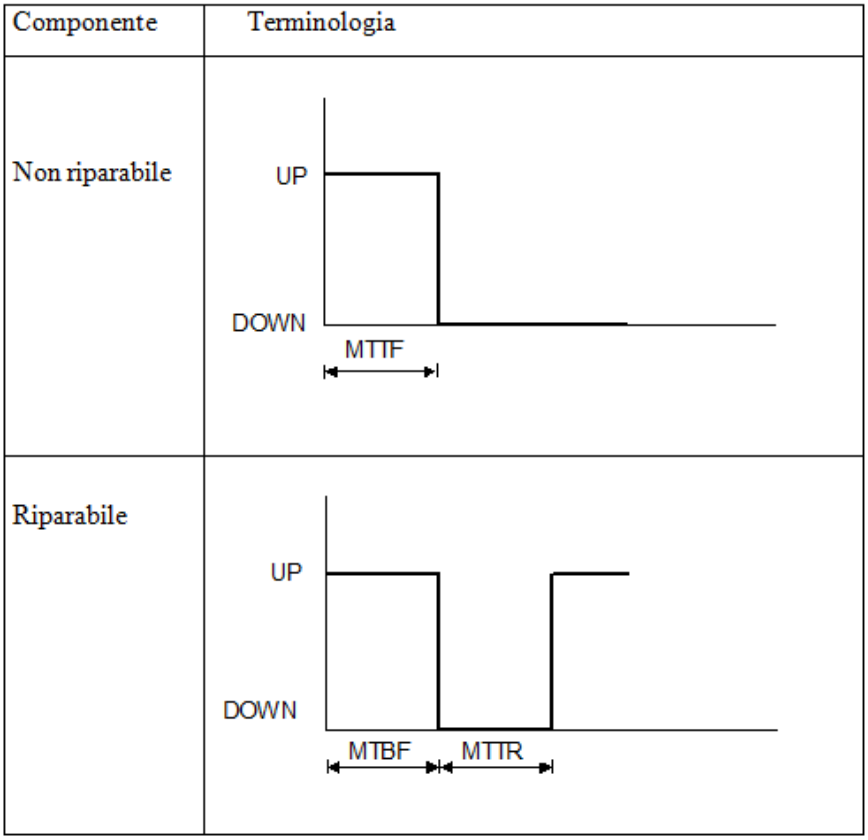


Figura A.7: MTTF (o MTBF) e MTTR

CASO PARTICOLARE:

Se il tasso di riparazione è costante $z(t) = cost = \mu$ si ha: $MTTR = \int_0^\infty e^{-\mu t} dt = -\left[\frac{e^{-\mu t}}{\mu}\right]_0^\infty = \frac{1}{\mu}$. Sostituendo il valore del tempo t = MTTR = $\frac{1}{\mu}$ alla relazione della probabilità di riparazione per componenti con tasso di riparazione costante si che ha M(MTTR) = $1 - e^{-1}$ = 0,63, cioè la probabilità che il componente venga riparato entro un tempo pari al MTTR è pari a circa il 63%.

PROCESSO ALEATORIO

Si chiama processo aleatorio una corrispondenza *x(t, ω)* tra le funzioni reali di una variabile indipendente t (normalmente il tempo) e gli elementi *ω* di uno spazio campione (Figura A.8). L'insieme delle possibili funzioni è indicato con X(t). Sia la funzione che il tempo possono assumere valori continui o discreti. Si hanno dunque quattro tipi di processi:

- Processi Aleatori **tempo-continui** a **valori continui**;
- Processi Aleatori **tempo-continui** a **valori discreti**;
- Processi Aleatori **tempo-discreti** a **valori continui**;
- Processi Aleatori **tempo-discreti** a **valori discreti**.

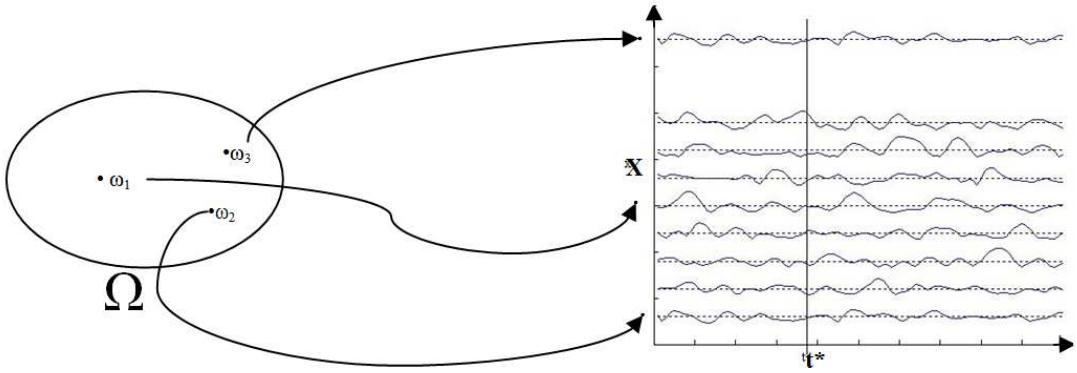


Figura A.8: Un processo stocastico, le sue realizzazioni e le sue n V.A. al tempo t*

Campionando il processo in n istanti di tempo $t_1, t_2, \ldots, t_n$ si ottengono n variabili aleatorie, la cui distribuzione congiunta è:

$$F(x_{1,}\, x_{2,\ldots,}\, x_{n,}\, ;\, t_{1,}t_{2,\ldots,}\, t_{n}) = P\{(X(t_1) \leq x_1,\, X(t_2) \leq x_2\, ,\, X(t_n) \leq x_n)\} \qquad (E.A.40)$$

Invece, fissato t all'istante t_1, il processo stocastico diventa $X(\omega_i,t_1)$. Abbiamo, dunque, n valori (tanti quanti la cardinalità dell'insieme S della classe degli eventi) corrispondenti ad un risultato dello spazio campione. Quindi, il valore del processo in un dato istante è una sola variabile aleatoria.

Invece, conoscendo il risultato dell'esperimento ω, ad esempio ω_1, selezioniamo quella tra le varie funzioni campione che si è realizzata in una data prova; non c'è alcuna aleatorietà e il processo diventa a posteriori un segnale determinato $X(\omega_1, t)$, cioè la funzione campione $x_1(t)$. In questo caso si parla di una realizzazione del processo.

Il processo stocastico è caratterizzato dalle relazioni fra le V.A. che lo compongono. A rigore, per conoscere un processo stocastico, è necessario conoscere tutte le distribuzioni di probabilità congiunta.

ANALISI DI MARKOV

L'affidabilità dei sistemi può essere studiata anche attraverso un'analisi di tipo markoviano; essa consente di superare alcuni dei limiti di altre metodologie, tra cui quello di non potere considerare guasti statisticamente dipendenti (Birolini, 2003), (Modarres et al., 1999), (Aven & Jensen, 1998).

DIAGRAMMI DEGLI STATI E PROCESSI STOCASTICI D-D E D-C

Un processo stocastico può essere definito come una successione di valori distinti assunti da una variabile casuale, $\{X_t\}$ con $t \in \mathbb{N}^+$.

Ogni valore X_t, appartiene ad un insieme finito di possibili valori che prendono il nome di stati del sistema e che possono essere di tipo sia qualitativo sia quantitativo; t è invece la variabile evolutiva che normalmente rappresenta il tempo.

I processi stocastici possono essere analizzati attraverso i modelli di Markov e si possono rappresentare con dei grafi detti **diagrammi degli stati** costituiti da eventi e da probabilità di transizione da un evento ad un altro.

In particolare si hanno due tipi di modelli:

- D-D: stati discreti con variabile temporale discreta (CATENE DI MARKOV)
- D-C: stati discreti con variabile temporale continua (PROCESSI MARKOVIANI)

Nelle analisi affidabilistiche vi sono applicazioni sia ai sistemi D-D sia ai sistemi D-C, come mostra schematicamente la Tabella A.2.

Tabella A.2: Applicazioni dei metodi di Markov

Variabile t	**Modello**	**Applicazioni**
Discreta	D-D	analisi di guasto
Continua	D-C	calcolo di R, A

In Figura A.9 è mostrato un esempio di una catena di Markov. Il modello viene realizzato per studiare un semplice caso studio. I dati del modello prevedono che:

- la domanda settimanale di un prodotto possa variare tra 0 e 3 unità;
- a secondo della giacenza X_i del sabato sera (che può assumere valori discreti [0÷3]), il venditore si rifornisce con 3 unità se la giacenza in magazzino è nulla ($X_i = 0$) oppure non si rifornisce se è presente in giacenza almeno un'unità;
- si suppone che la consegna in magazzino avvenga sempre prima di lunedì.

La situazione può essere schematizzata con il diagramma degli stati di Figura A.9, seguendo il seguente formalismo:

X → giacenza a fine settimana;

D → domanda settimanale;

A → acquisto (solo in caso di esaurimento scorte).

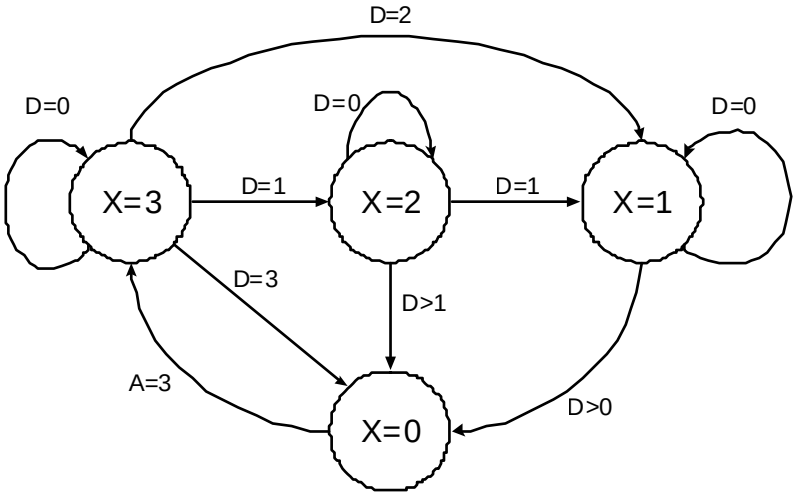


Figura A.9: Diagramma per il modello delle vendite settimanali (vedi sopra)

CATENE DI MARKOV (D-D)

Le catene di Markov (Gallager, 1996) costituiscono un processo stocastico, cioè una successione di variabili aleatorie finite, con le seguenti proprietà:

1. lo spazio degli stati è costituito da un numero finito di stati (X_1, X_2, ..., X_n);
2. le transizioni avvengono (o comunque sono osservate) solo ad intervalli discreti del tempo;
3. ogni stato dipende esclusivamente dallo stato immediatamente precedente e non da tutta la "storia" precedente: a ciascuna coppia di stati (X_i, X_j) viene associato un valore p_{ij} che è la probabilità che lo stato X_j si verifichi immediatamente dopo X_i.

Le probabilità di transizione p_{ij} possono essere ordinate in forma matriciale, ottenendo la cosiddetta **matrice di transizione P**:

$$P = \begin{pmatrix} p_{11} & p_{12} & \cdots & p_{1n} \\ p_{21} & p_{22} & \cdots & p_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ p_{n1} & p_{n2} & \cdots & p_{nn} \end{pmatrix}$$

nella quale la i-esima riga contiene le probabilità che il sistema evolva dallo stato X_i verso gli altri possibili stati (incluso lo stesso stato X_i); perciò, ogni riga è un vettore delle probabilità e la matrice di transizione P di una catena markoviana è una matrice stocastica.

PROCESSI MARKOVIANI (MODELLO D-C)

Diversamente dalle catene di Markov, si considerano stati discreti e intervalli di tempo continui (dt). Si applicano le stesse ipotesi del caso D-D ma diversamente dalle probabilità di transizione, il passaggio da uno stato ad un altro viene regolato da funzioni del tempo (Nielses, 2009).

In applicazioni di tipo affidabilistico (Birolini, 2003), (Pham, 2003) gli stati corrispondono alle condizioni operative del sistema in esame (guasto, funzionante, ...) e le transizioni da uno stato all'altro corrispondono al numero di guasti o di riparazioni che avvengono nel sistema per unità di tempo. Se questi eventi avvengono con tassi di guasto e di riparazione costanti (il che significa che stiamo considerando la sola "vita utile" del nostro sistema) il processo di Markov corrispondente prende comunemente il nome di processo di Poisson e la matrice di transizione prende il nome di matrice dei generatori infinitesimali (Q) (Trivedi, 2002).

Anche nel modello D-C, lo stato in cui il sistema viene a trovarsi in un certo istante dipende solo da quello immediatamente (infinitesimamente) precedente e non dall'intera storia del sistema.

APPENDICE B

COMPORTAMENTO AFFIDABILISTICO DELLE PORTE DINAMICHE E RELAZIONE CON I MODELLI NELLO SPAZIO DEGLI STATI

In questa sezione dell'appendice vengono chiariti alcuni comportamenti tipici delle porte dinamiche che si possono presentare nella modellazione dei DFT.

PORTA PAND

Le porte PAND (Priority-AND) possono essere usate per modellare la logica di accadimento di un evento che deve verificarsi solo in seguito ad una ben precisa sequenza ordinata di eventi iniziatori. Se tale sequenza non viene rispettata, la porta PAND commuta nel senso opposto.

Da un punto di vista affidabilistico, le PAND sono la corretta alternativa alle porte AND degli SFT, per la modellazione delle logiche di prevenzione e allarme dei sistemi di sicurezza di un generico sistema.

La porta PAND commuta verso il valore booleano vero se e solo se tutti gli ingressi alla porta accadono nell'ordine che va da sinistra a destra.

L'uscita di una porta PAND può essere sia un Top Event che un evento intermedio.

Gli ingressi possono essere eventi Iniziatori (BE, UE, EE), nonché uscite proveniente dalle altre porte sia statiche che dinamiche.

In Figura B.1 è mostrata una porta PAND a 2 ingressi e la sua rappresentazione nello spazio degli stati mediante una GSMP (Younes & Simmons, 2004), (Glynn, 1989). Le funzioni $R_A(t)$, $R_B(t)$, $F_A(t)$ ed $F_B(t)$ rappresentano rispettivamente le funzioni di distribuzione associate alle transizioni che riguardano la riparazione (R(t)) e il guasto (F(t)) per gli eventi A e B.

La tavola della verità per la PAND a 2 ingressi è mostrata in Tabella B.1. La porta emette il valore booleano vero solo se A e B sono entrambe vere e, inoltre, il tempo di occorrenza di A è inferiore a quello di B, con $T_1 < T_2$. Tutte le altre combinazioni danno luogo ad un'uscita falsa.

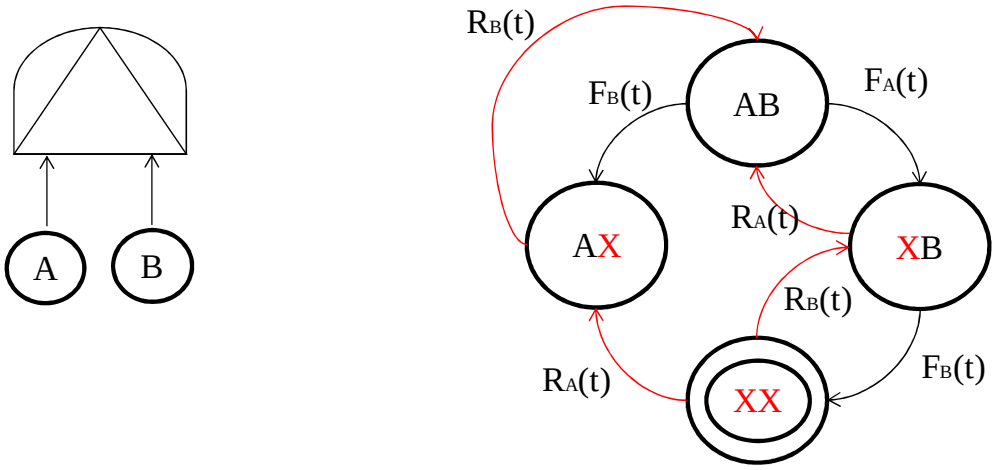


Figura B.1: PAND a 2 ingressi e rappresentazione nello spazio degli stati

Tabella B.1: tavola della verità per una PAND a 2 ingressi

A	B	Uscita
T_1	T_2	Vero
T_2	T_1	Falso
Vero	Falso	Falso
Falso	Vero	Falso
Falso	Falso	Falso

L'equivalente CTMC si ottiene sostituendo ad ogni funzione F(t) e R(t) il corrispondente tasso di transizione costante di guasto λ e di riparazione μ.

Affidabilità: tradizionalmente, questa misura viene calcolata ignorando tutte le transizioni che riguardano la riparazione dei componenti. In questo caso, la soluzione in forma chiusa per la PAND a 2 ingressi risulta essere:

$$\boldsymbol{F(t) = P(XX) = \frac{\lambda_B}{\lambda_A+\lambda_B}\left(e^{-(\lambda_A+\lambda_B)t} - 1\right) - e^{-\lambda_B t} + 1} \qquad \textit{(B.1)}$$

L'aspetto interessante che risulta dal calcolo di questa misura è il valore di regime che si può osservare. Non appena i modi esponenziali, al tendere di $t \to \infty$, diventano trascurabili l'inaffidabilità della porta si assesta al valore di regime

$$F_{reg}(t) = \lim_{t\to\infty} P(XX) = 1 - \frac{\lambda_B}{\lambda_A+\lambda_B} \qquad (B.2)$$

Il risultato è spiegabile perché, contrariamente ad una porta AND, l'assenza della transizione di rottura dallo stato 'AX' verso lo stato 'XX' fa si che lo stato 'AX' sia uno stato assorbente non di guasto. La presenza di questi stati, conosciuti con la denominazione di stati *safe*, rendono il sistema molto affidabile. Tuttavia la loro consistenza, in termini di modello affidabilistico, deve essere valutata con criterio. Infatti, nella realtà uno stato di tipo safe (di mancato pericolo, di mancato guasto, ecc.) è comunque uno stato in cui la normale operatività del sistema è alterata e per cui non ha senso il permanere del sistema in questo stato. Per questo motivo, la porta PAND dovrebbe prevedere una transizione da questo stato verso quello che riporta il sistema in una condizione di normale operatività.

Disponibilità: si è verificato che la forma chiusa di una semplice PAND a 2 ingressi ha un'espressione già molto complicata.

La logica di funzionamento della porta può essere discussa verificando le transizioni del diagramma nello spazio degli stati (Figura B.1). Emergono comportamenti particolarmente interessanti ai fini delle valutazioni di rischio.

Trattandosi di disponibilità del sistema, si supponga che il sistema sia in uno stato di indisponibilità (lo stato 'XX') al tempo T_1. Dal diagramma degli stati, da questo stato 'XX', sono ammissibili due transizioni di riparazione dei componenti verso gli stati 'AX' o 'XB'. Questi due stati sono entrambi di funzionamento per il sistema, per cui concorrono ad aumentare la disponibilità globale. Tuttavia, data la logica della porta PAND, dallo stato 'AX' non è possibile raggiungere lo stato 'XX'. Dunque, una volta tornato sullo stato 'AX', il sistema non può più transitare verso 'XX' senza prima ritornare allo stato iniziale 'AB', fintanto cioè che il secondo componente (B) non venga riparato riportando il sistema come nuovo.

Un esempio impiantistico che mostra come potrebbe essere modellata correttamente una PAND è quello di Figura B.2 (sistema di controllo del fluido di un reattore).

Questo semplice esempio spiega l'importanza della fase di modellazione per ciò che riguarda la congruenza tra la logica implementata nel modello di rischio, mediante le porte dinamiche, e quella reale del processo studiato.

L'alto o il basso livello (TE del DFT) si registra soltanto se un'anomalia alla valvola non è rilevata dal sensore di livello. In termini di CTMC lo schema è sintetizzato in Figura B.3, dove A = Sensore Livello, B = Valvola. Affinché il livello di fluido sia troppo alto/basso (dunque non congruo con quello di processo) deve accadere per primo il guasto del sensore di livello (che regola l'apertura e la chiusura automatica della valvola). In questo caso lo schema nello spazio degli stati per la disponibilità potrebbe essere come quello in Figura B.3 in cui dallo stato 'XX' è possibile solo la transizione di riparazione che riporta il sistema allo stato iniziale 'AB'. Infatti, da un punto di vista manutentivo si assume che l'avvenimento del TE solleciti il team della manutenzione a riportare il sistema allo stato AB facendo un'ispezione sul sensore di livello ed aggiustando contemporaneamente valvola e sensore.

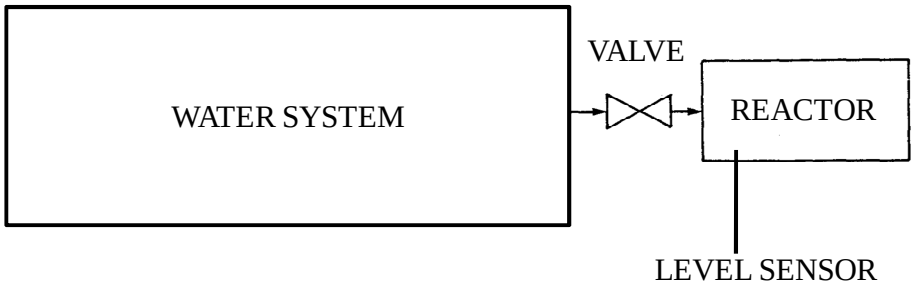


Figura B.2 : Sistema di controllo del livello di fluido nel reattore

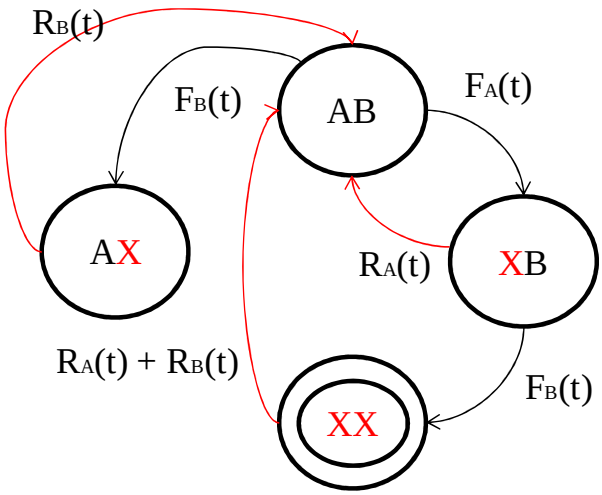


Figura B.3: Disponibilità del sistema di Figura B.2 (GSMP)

Prima evenienza del TE: in termini di modellazione del rischio, per l'esempio di Figura B.2 ha più senso il calcolo della prima evenienza del TE. Infatti, l'insorgere di una condizione anomala nel livello di fluido può essere causa di un processo irreversibile (dallo stato 'XX' non si torna più indietro) il cui rischio va contemplato nel rapporto di sicurezza. In questi casi, lo schema che ha più senso per la porta PAND è quello di Figura B.3 privato della transizione di riparazione dallo stato 'XX'. In tale caso la reversibilità del processo è ammessa fintanto che il TE non sia avvenuto, quando ancora è possibile intervenire sulle parti del sistema per riportarlo in condizioni di normale operatività.

PORTA SPARE

La porta SPARE viene utilizzata per modellare sistemi con parti di ricambio nella logica parallela cold, warm e hot stand-by.

Nella porta SPARE, il primo ingresso (il più a sinistra), rappresenta il componente attivo principale; tutti gli altri ingressi sono i suoi possibili componenti di sostituzione che, come detto, possono inizialmente trovarsi in una modalità standby.

Gli ingressi della porta SPARE che fungono da ricambio possono essere usati all'ingresso di altre SPARE, emulando il comportamento di un sistema in cui uno stesso componente possa, all'occorrenza, sostituire altri componenti primari attivi.

In Tabella B.2 è mostrata la tavola della verità della porta SPARE a 2 ingressi.

Tabella B.2: tavola della verità per una SPARE a 2 ingressi

A	B	Uscita
T	T	Vero
T	F	Falso
F	T	Falso
Falso	Falso	Falso

Un esempio di modello con porta SPARE potrebbe il sistema "ruote-ruota di scorta" di un'automobile (vedi Figura B.4): gli ingressi di ogni porta SPARE sono le quattro ruote di serie (R_i, i=1,..,4) e la ruota di scorta (R_s). La modellazione prevede il guasto

dell'automobile se una delle porte SPARE risulta vera. Ciò viene realizzato attraverso una configurazione OR che prende in ingresso le uscite delle porte SPARE.

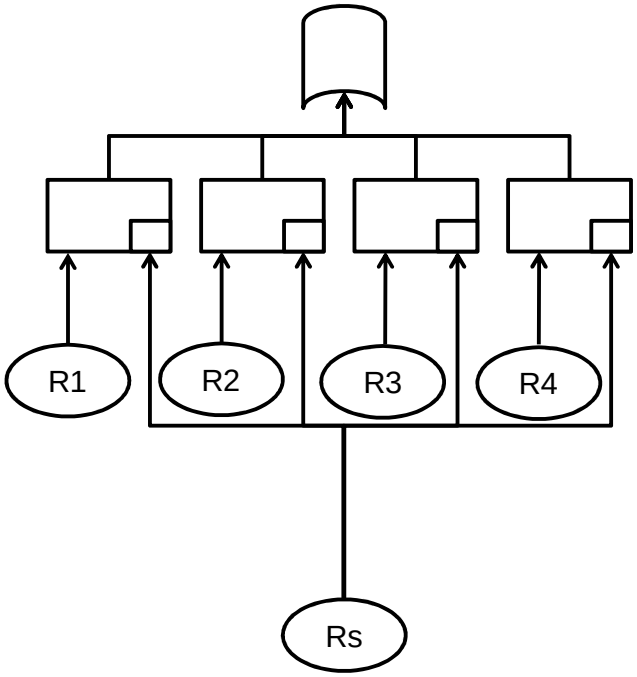


Figura B.4: modellazione mediante SPARE di un cambio ruota di un'automobile

Dopo il guasto del componente primario, il componente di sostituzione più a sinistra (se non usato da altre SPARE) viene messo in modalità attiva rimpiazzando il componente principale. Se tutte le unità di sostituzione sono state già usate da altre porte SPARE, la SPARE in questione sarà valutata fallita.

La modellazione di un componente cold, warm oppure hot stand-by viene realizzata mediante un fattore moltiplicativo, il fattore di latenza, compreso tra 0 e 1:

- se tale fattore è 1, il componente è in modalità hot stand-by;
- se è compreso tra 0 e 1, il componente di ricambio risulta in modalità warm fino a quando diventa utilizzabile. A quel punto il fattore diventa unitario;
- se tale fattore è 0, allora il componente di sostituzione è in modalità cold e fintanto che non viene messo in funzione non può subire alcun guasto.

In Figura B.5 è mostrata la GSMP di una SPARE a due ingressi, con un primario (A) ed un componente di sostituzione (S).

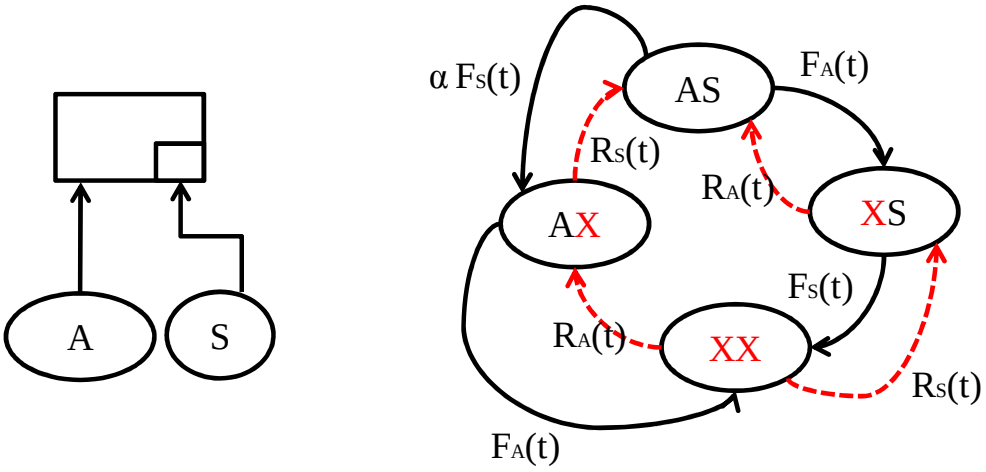


Figura B.5: SPARE a 2 ingressi e corrispondente GSMP. Il parametro α è il fattore di latenza

Affidabilità: l'affidabilità di un modello con la SPARE viene calcolata per sistemi i cui componenti non sono riparabili. In Figura B.5, la GSMP corrispondente va privata delle transizioni di riparazione (tratteggiate). Il fattore α di latenza modella il tipo di configurazione stand-by.

Se si considera la corrispondente CTMC per il diagramma degli stati di Figura A.5, la formula dell'inaffidabilità in forma chiusa è:

$$\boldsymbol{F(t) = P(XX) = \frac{\lambda_A}{\lambda_A+\lambda_S(\alpha-1)}[e^{-(\lambda_A+\alpha\lambda_S)t} - e^{-\lambda_S t}] - e^{-\lambda_A t} + 1} \quad \textit{(E.B.2)}$$

Disponibilità: rispetto alla PAND, la modellazione della disponibilità con una porta SPARE non soffre di nessun equivoco. Infatti, se almeno uno dei due componenti è attivo (o torna ad essere funzionante dopo un guasto) il sistema generale risulta disponibile.

Prima evenienza del TE: anche per questa misura la porta SPARE risulta inequivocabile. Le transizioni di riparazione dallo stato 'XX' vengono elise, in modo che 'XX' risulti assorbente.

PORTA SEQ

La porta SEQ forza gli eventi ad accadere in un preciso ordine. Gli eventi in ingresso alla porta sono costretti a verificarsi in un ordine diretto da sinistra verso destra rispetto a come appaiono nel DFT: ciò significa che un evento alla destra di un altro può avvenire soltanto immediatamente dopo l'evento alla sua sinistra.

La differenza tra una porta SEQ e una porta PAND sta nel fatto che nella prima non sono ammesse disposizioni temporalmente disordinate dell'ordine di occorrenza degli eventi, mentre nella seconda ciò è consentito e dando luogo a commutazione della porta sia vere che non. In altre parole, mentre la SEQ forza la dinamica ad accadere in un ordine preciso, la PAND effettua invece un controllo di questa dinamica rilevando l'ordine temporale con cui gli eventi si susseguono.

Il primo ingresso (più a sinistra) ad una SEQ può essere un evento BE, EE, UE o l'uscita di ogni porta statica o cancello dinamico. Tutti gli altri ingressi possono solo essere eventi di tipo BE, EE o UE.

Una porta SEQ viene spesso utilizzata per modellare la degradazione progressiva di un sistema. Per esempio, si consideri un processo industriale per il taglio di profili metallici la cui affidabilità è basata sulla qualità della lama. Inizialmente, ci possono essere degradazioni minime alla lama in modo che il sistema risulti ancora funzionante e in buone condizioni. Progressivamente, possono verificarsi delle degradazioni alla lama tali che i tagli risultino meno efficaci, pur mantenendo il sistema è funzionante, quindi affidabile. Infine, il momento in cui tali degradi rendono il taglio della lama inaccettabile, il sistema è considerato danneggiato.

In Tabella B.3 è riportata la tavola della verità della SEQ a 3 ingressi. L'unica combinazione vera ammissibile è quella che vede il verificarsi della sequenza A-B-C secondo i tempi $T_1<T_2<T_3$. Tutte le altre possibili rappresentazioni non sono ammissibili. In Figura B.6 è riportata la corrispondente GSMP della SEQ a 3 ingressi.

L’equivalente CTMC si ottiene sostituendo ad ogni funzione F(t) e R(t) il corrispondente tasso di transizione costante λ e μ.

Tabella B.3: Tavola della verita della SEQ a 3 ingressi

A	B	C	Uscita
Falso	Falso	Falso	Falso
Falso	Falso	T	Impossibile
Falso	T	Falso	Impossibile
Falso	T	T	Impossibile
T_1	Falso	Falso	Falso
T_1	Falso	T	Impossibile
T_1	T_2	Falso	Falso
T_1	T_2	T_3	Vero

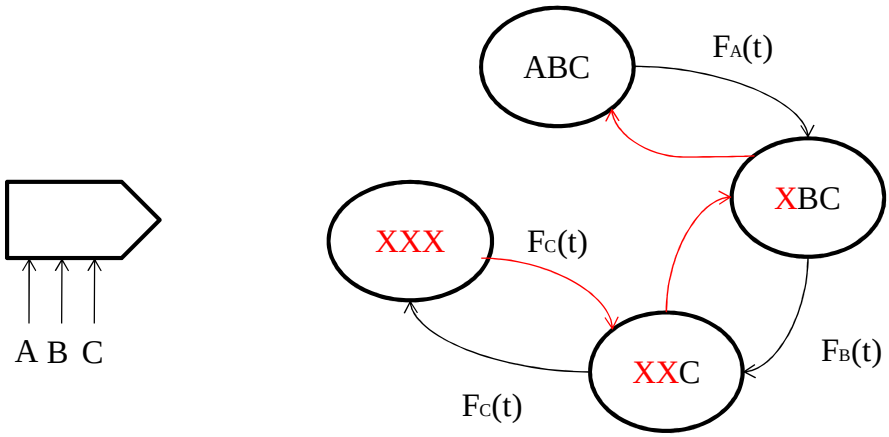


Figura B.6: SEQ a 3 ingressi e GSMP corrispondente

Affidabilità: il calcolo dell'affidabilità viene effettuato trascurando le transizioni di riparazione. In questo caso, la soluzione in forma chiusa per la SEQ a 3 ingressi risulta:

$$\boldsymbol{F(t) = P(XXX) = \frac{\lambda_A\lambda_C}{(\lambda_A-\lambda_B)(\lambda_B-\lambda_C)}e^{-\lambda_B t} - \frac{\lambda_B\lambda_C}{(\lambda_A-\lambda_B)(\lambda_A-\lambda_C)}e^{-\lambda_A t} - \frac{\lambda_A\lambda_B}{(\lambda_A-\lambda_C)(\lambda_B-\lambda_C)}e^{-\lambda_C t} + 1} \qquad (E.B.3)$$

Disponibilità: tutti gli stati eccetto il 'XXX' sono di disponibilità per il sistema. La SEQ non presenta ambiguità rispetto alla valutazione della disponibilità. Tuttavia, per sistemi reali può avere senso l'utilizzo di una sola transizione di riparazione a

partire dallo stato finale che riporti il sistema allo stato iniziale, piuttosto che allo stato di degrado immediatamente precedente a quello di guasto.

Prima evenienza del TE: la modellazione dell'occorrenza di un evento mediante una SEQ non presenta ambiguità logiche e l'unica transizione da elidere è (Figura B.6) la $F_C(t)$.

PORTA FDEP

La porta FDEP accetta due tipi di ingresso (Figura B.7): l'evento principale (o scatenante, P) e gli eventi dipendenti (D1, D2, ...Dn). Questa porta forza gli eventi dipendenti ad accadere secondo l'ordine con cui sono connesse alla porta non appena l'evento scatenante avviene.

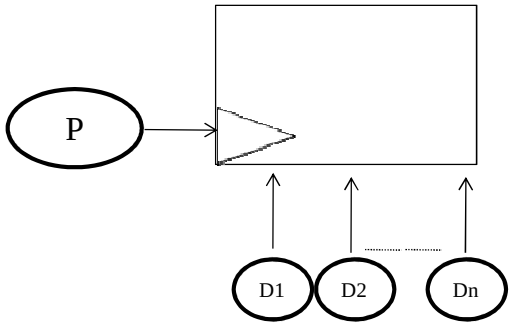


Figura B.7: FDEP con ingresso scatenante P ed n ingressi dipendenti

La FDEP ha un'uscita fittizia che non si propaga attraverso il DFT.

Gli ingressi di tipo principale possono essere eventi iniziatori (BE, EE, UE) o uscite di altre porte; gli eventi dipendenti sono eventi che hanno connessioni con altre porte del DFT e sono anch'essi appartenenti all'insieme degli eventi iniziatori.

Da un punto di vista dell'analisi di rischio, la FDEP aumenta la probabilità propria di occorrenza di un evento facendolo dipendere da quella dell'evento scatenante.

La modellazione con una FDEP semplifica il processo di costruzione di un DFT. Infatti, un DFT con una porta FDEP corrisponde esattamente ad un DFT in cui la porta FDEP viene sostituita con tante porte OR (Figura B.8) quanti sono gli ingressi dipendenti. Queste porte OR accettano in ingresso l'evento scatenante e i singoli ingressi secondari; l'uscita di ogni OR va in ingresso alla porta che riceveva (nel modello con la FDEP) in ingresso l'evento dipendente.

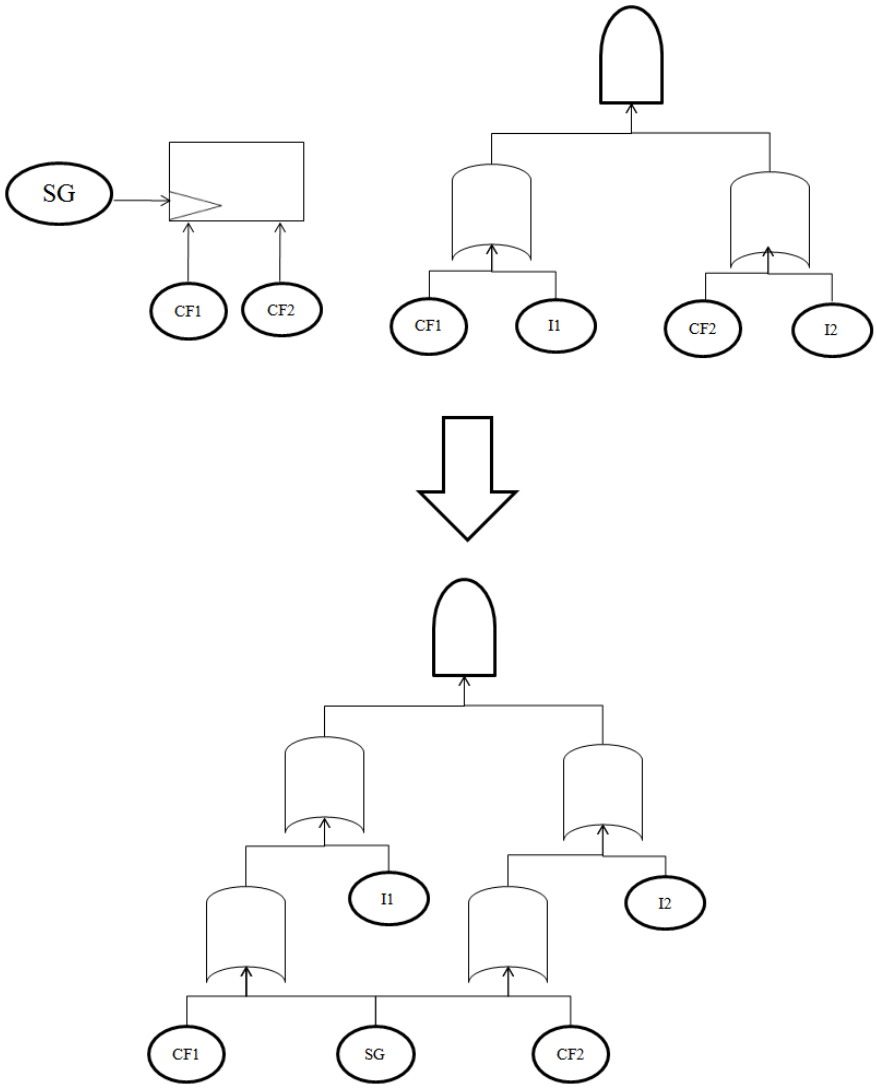


Figura B.8: FDEP con ingresso scatenante SG, 2 ingressi dipendenti (CF1 e CF2) e la modellazione equivalente mediante le OR

APPENDICE C

CODIFICA DEL MODELLO DI SENSOR NETWORK

Di seguito sono presentati i due codici in SHARPE (Sahner & Trivedi, 1987), (Pan, 2001) per il calcolo delle IMs tradizionali (**codice 1**) e (**codice 2**) mediante la tecnica FT (Fricks & Trivedi, 2003) della Sensor Network (Kim et al., 2010 in publishing).

Ogni **linea del codice** in SHARPE rappresenta un'istruzione. Ogni istruzione va terminata con il carattere '\n' (new line), andando a capo.
Il **commento** si produce apponendo ad ogni linea il carattere '%'.
Le parola chiave:

1. '**factor**' abilita l'algoritmo per il calcolo degli sdp mediante fattorizzazione.
2. '**bind**' associa alla variabile il valore numerico immediatamente alle destra, chiudendo l'assegnazione con la parola chiave 'end' (quando sono presenti più associazioni).
3. **'markov', 'ftree'** cominciano una sequenza di istruzioni per descrivere un sistema mediante Markov Chain o un Fault Tree.

Per ogni altra informazione sul software SHARPE, si consiglia la visita del sito http://sharpe.pratt.duke.edu/.

CODICE 1

```
format 8
factor on

%PARAMETRI DELLA SENSOR NETWORK

bind
beta_wkp   1/100
lam_bd   1/8760
lam_tra   1/10000
lam_v   1/100
lam_fp   1/720
lam_mic   1/10000
```

```
lam_pwr   1/10000
Cs   0.6
Cr   0.95
lam_f1   1/480
r   1/1200
lam_uc   60
alpha_slp   1/100000000
lam_mem   1/10000
lam_a   1/24
lam_app   1/100000
mu_rj   20
mu_s   6
delta_s   30
lam_sen   1/10000
end

% CTMC DEL SISTEMA POWER
markov power
SLP N beta_wkp
N F lam_pwr
N SLP alpha_slp
N BD lam_bd
end
* Initial Probabilities defined:
SLP init_power_SLP
N init_power_N
BD init_power_BD
F init_power_F
end

* Initial Probailities assigned:
bind
  init_power_SLP 0
  init_power_N 0
  init_power_BD 0
  init_power_F 0
end

* Initial Probability: init
bind
  init_power_SLP  0
  init_power_F  0
  init_power_BD  0
  init_power_N  1
end

% CTMC DEL SISTEMA SENSORE
```

```
markov sensor
UP F lam_sen
end
* Initial Probabilities defined:
UP init_sensor_UP
F init_sensor_F
end

* Initial Probailities assigned:
bind
  init_sensor_UP 0
  init_sensor_F 0
end

* Initial Probability: init
bind
  init_sensor_UP  1
  init_sensor_F  0
end

% CTMC DEL SISTEMA  MICRO
markov micro
UP F lam_mic
end
* Initial Probabilities defined:
UP init_micro_UP
F init_micro_F
end

* Initial Probailities assigned:
bind
  init_micro_UP 0
  init_micro_F 0
end

* Initial Probability: init
bind
  init_micro_UP  1
  init_micro_F  0
end

% CTMC DEL SISTEMA MEMORY
markov memory
UP F lam_mem
end
* Initial Probabilities defined:
UP init_memory_UP
```

```
F init_memory_F
end

* Initial Probailities assigned:
bind
  init_memory_UP 0
  init_memory_F 0
end

* Initial Probability: init
bind
  init_memory_UP  1
  init_memory_F  0
end

% CTMC DEL SISTEMA TRANSCEIVER
markov trans
UP F lam_tra
end
* Initial Probabilities defined:
UP init_trans_UP
F init_trans_F
end

* Initial Probailities assigned:
bind
  init_trans_UP 0
  init_trans_F 0
end


* Initial Probability: init
bind
  init_trans_UP  1
  init_trans_F  0
end

% CTMC DEL SISTEMA TINYOS
markov tinyos
G V lam_v
G FP lam_fp
V A lam_a
FP R r
FP F lam_f1
A D Cs*delta_s
A UC (1-Cs)*lam_uc
D G mu_s
```

```
R F (1-Cr)*mu_rj
R G Cr*mu_rj
end
* Initial Probabilities defined:
G init_tinyos_G
V init_tinyos_V
FP init_tinyos_FP
A init_tinyos_A
D init_tinyos_D
UC init_tinyos_UC
R init_tinyos_R
F init_tinyos_F
end

* Initial Probailities assigned:
bind
  init_tinyos_G 0
  init_tinyos_V 0
  init_tinyos_FP 0
  init_tinyos_A 0
  init_tinyos_D 0
  init_tinyos_UC 0
  init_tinyos_R 0
  init_tinyos_F 0
end

* Initial Probability: init
bind
  init_tinyos_A  0
  init_tinyos_V  0
  init_tinyos_UC  0
  init_tinyos_FP  0
  init_tinyos_R  0
  init_tinyos_G  1
  init_tinyos_F  0
  init_tinyos_D  0
end

markov app
UP F lam_app
* Reward configuration defined:
reward
UP rew_app_UP
F rew_app_F
end
* Initial Probabilities defined:
UP init_app_UP
```

```
F init_app_F
end

* Reward configuration assigned:
bind
  rew_app_UP 0
  rew_app_F 0
end

* Initial Probailities assigned:
bind
  init_app_UP 0
  init_app_F 0
end

* Initial Probability: init
bind
  init_app_UP  1
  init_app_F  0
end

% FAULT TREE STATICO DELLA SENSOR NETWORK
% IN INGRESSO SONO PRESENTI ANCHE I SISTEMI DELLE CTMC

ftree singsen(t)
basic pow1 prob(tvalue(t;power))
basic sen1 prob(tvalue(t;sensor))
basic mic1 prob(tvalue(t;micro))
basic mem1 prob(tvalue(t;memory))
basic tra1 prob(tvalue(t;trans))
basic tin1 prob(tvalue(t;tinyos))
basic app1 prob(tvalue(t;app))
or or3  tin1 app1
or or1  pow1 sen1 mic1 mem1 tra1
or or0  or1 or3
end

echo**********************************
********************************** ****
echo *********  Outputs asked for  the model: singsen **********

% VALUTAZIONI DELL'AFFIDABILITA' E DELLE IMs

func Reliability(t)  1-tvalue(t;singsen;t)
loop t,0,2500,100
expr Reliability(t)
expr 1-tvalue(t;power)
```

```
expr 1-tvalue(t;sensor)
expr 1-tvalue(t;micro)
expr 1-tvalue(t;memory)
expr 1-tvalue(t;trans)
expr 1-tvalue(t;tinyos)
expr 1-tvalue(t;app)

expr bimpt(t;singsen,pow1;t)
expr bimpt(t;singsen,sen1;t)
expr bimpt(t;singsen,mic1;t)
expr bimpt(t;singsen,mem1;t)
expr bimpt(t;singsen,tra1;t)
expr bimpt(t;singsen,tin1;t)
expr bimpt(t;singsen,app1;t)
expr cimpt(t;singsen,pow1;t)
expr cimpt(t;singsen,sen1;t)
expr cimpt(t;singsen,mic1;t)
expr cimpt(t;singsen,mem1;t)
expr cimpt(t;singsen,tra1;t)
expr cimpt(t;singsen,tin1;t)
expr cimpt(t;singsen,app1;t)
end


end
```

CODICE 2

Per la codifica del secondo script si sono implementate funzioni di supporto che servono per effettuare l'algoritmo che calcola i parametri tipici delle FTM, misura non implementata nel motore di SHARPE.

```
format 8
factor on

bind
beta_wkp   1/100
lam_bd   1/8760
lam_tra   1/10000
lam_v   1/100
lam_fp   1/720
lam_mic   1/10000
lam_pwr   1/10000
Cs   0.6
Cr   0.95
lam_f1   1/480
```

```
r   1/1200
lam_uc   60
alpha_slp   1/100000000
lam_mem   1/10000
lam_a   1/24
lam_app   1/100000
mu_rj   20
mu_s   6
delta_s   30
lam_sen   1/10000
end

*CODIFICA FUNZIONI DI SUPPORTO

func unreal_comp(l,t)
1-^(-l*t)
end

func wrapFT(real_val,j,k,c)
if(c==j)
 if(k==1)
 0
 else
 1
 end
else
real_val
end
end

func  Store(k,unreal,unreal_temp)
if(k==0)
unreal
else
unreal_temp
end
end

func  StoreFT(k,FT,FT_temp,j)
if(j==k)
FT
else
FT_temp
end
end

func  computeRRW(k,num,RRW_temp,j ,c,unreliab)
```

```
if(unreliab==0)
0
else
if(j==c)
 if(k==1)
 num/unreliab
 else
 RRW_temp
 end
else
RRW_temp
end
end
end


func  computeRAW(k,num,RAW_temp,j ,c,unreliab)
if(unreliab==0)
0
else
if(j==c)
 if(k==0)
 num/unreliab
 else
 RAW_temp
 end
else
RAW_temp
end
end
end

func computeCFT(x,unrealiability)
if (unreliability==0 )
0
else
x/unreliability
end
end

func computeDIM(x,unrealiability)
if (unreliability==0 )
0
else
x/unreliability
end
end
```

```
**Modify:depends on the  number of component
*in the MarkovChain plus  the standard configuration
*n_state:4
*n_component:2 ->  Battery(1), Power(2)

func  assign_rew_power(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 1 (power),  n_component(j) = 2: 1,2
**first component of c=1:
*j=1, battery
 if(j==1)
  if(k==0)
   if(state==0)
   1
   elseif(state==1)
   1
   elseif(state==2)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   elseif(state==1)
   1
   elseif(state==2)
   0
   else
   0
   end
  end
*j=2 sistema in se
  elseif(j==2)
   if(k==0)
   if(state==0)
   1
   elseif(state==1)
   1
   elseif(state==2)
   1
   else
   1
```

```
   end
  elseif(k==1)
   if(state==0)
   0
   elseif(state==1)
   0
   elseif(state==2)
   1
   else
   0
  end
*   chiusura k
  end
else
   if(state==0)
   0
   elseif(state==1)
   1
   elseif(state==2)
   1
   else
   0
   end
 end
end

func  assign_rew_sensor(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 2 (sensor),  n_component(j) = 1: 3
**first component of c=2:
*j=3, sensor
 if(j==3)
  if(k==0)
   if(state==0)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   else
   0
   end
```

```
*   chiusura k
  end
  else
   if(state==0)
   0
   else
   1
   end
 end
end

func  assign_rew_micro(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 3 (micro),  n_component(j) = 1: 4
**first component of c=3:
*j=4, micro
 if(j==4)
  if(k==0)
   if(state==0)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   else
   0
   end
*   chiusura k
  end
  else
   if(state==0)
   0
   else
   1
   end
 end
end

func  assign_rew_memory(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
```

```
*then components
***c = 4 (memory),  n_component(j) = 1: 5
**first component of c=4:
*j=5, memory
 if(j==5)
  if(k==0)
   if(state==0)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   else
   0
   end
*   chiusura k
  end
  else
   if(state==0)
   0
   else
   1
   end
 end
end

func  assign_rew_trans(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 5 (trans),  n_component(j) = 1: 6
**first component of c=5:
*j=5, trans
 if(j==6)
  if(k==0)
   if(state==0)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   else
```

```
   0
   end
*   chiusura k
  end
  else
   if(state==0)
   0
   else
   1
   end
 end
end

func  assign_rew_tynos(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 6 (tinyos),  n_component(j) = 2: 1,2
**first component of c=6:
*j=7, detective
 if(j==7)
  if(k==0)
   if(state==0)
   0
   elseif(state==1)
   0
   elseif(state==2)
   0
   elseif(state==3)
   1
   elseif(state==4)
   0
   elseif(state==5)
   1
   elseif(state==6)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   elseif(state==1)
   0
   elseif(state==2)
   0
```

```
  elseif(state==3)
  0
  elseif(state==4)
  0
  elseif(state==5)
  1
  elseif(state==6)
  0
  else
  0
  end
 end
*j=8 failure event
 elseif(j==8)
  if(k==0)
  if(state==0)
  0
  elseif(state==1)
  1
  elseif(state==2)
  1
  elseif(state==3)
  1
  elseif(state==4)
  1
  elseif(state==5)
  1
  elseif(state==6)
  1
  else
  1
  end
 elseif(k==1)
  if(state==0)
  0
  elseif(state==1)
  0
  elseif(state==2)
  0
  elseif(state==3)
  0
  elseif(state==4)
  0
  elseif(state==5)
  0
  elseif(state==6)
  1
```

```
   else
   0
   end
  end
*j = 9 rejuvenation mechanism
elseif(j==9)
  if(k==0)
   if(state==0)
   0
   elseif(state==1)
   0
   elseif(state==2)
   1
   elseif(state==3)
   0
   elseif(state==4)
   1
   elseif(state==5)
   1
   elseif(state==6)
   1
   else
   0
   end
* rejuvenation  succeeds
  elseif(k==1)
   if(state==0)
   0
   elseif(state==1)
   0
   elseif(state==2)
   0
   elseif(state==3)
   0
   elseif(state==4)
   0
   elseif(state==5)
   0
   elseif(state==6)
   1
   else
   0
   end
  end

else
   if(state==0)
```

```
   0
   elseif(state==1)
   0
   elseif(state==2)
   0
   elseif(state==3)
   0
   elseif(state==4)
   0
   elseif(state==5)
   1
   elseif(state==6)
   1
   else
   0
   end
  end
end

func  assign_rew_app(c,k,j,state)
*c= to call the right  subsystm
*j=the number of the  component on the DFT
*k= 0/1 (component of the  subsystem dw or up)
*poi passo ai componenti
***c = 7 (app),  n_component(j) = 1: 10
**first component of c=7:
*j=10, app
 if(j==10)
  if(k==0)
   if(state==0)
   1
   else
   1
   end
  elseif(k==1)
   if(state==0)
   0
   else
   0
   end
*   chiusura k
  end
  else
   if(state==0)
   0
   else
   1
```

```
   end
 end
end


markov power
SLP N beta_wkp
N F lam_pwr
N SLP alpha_slp
N BD lam_bd

reward
N rew_power_0
F rew_power_1
BD rew_power_2
SLP rew_power_3
end

* Initial Probabilities  defined:
SLP init_power_SLP
N init_power_N
BD init_power_BD
F init_power_F
end


* Initial Probailities  assigned:
bind
   init_power_SLP 0
   init_power_N 0
   init_power_BD 0
   init_power_F 0
end

* Initial Probability: init
bind
   init_power_SLP  0
   init_power_F  0
   init_power_BD  0
   init_power_N  1
end


markov sensor
UP F lam_sen
reward
UP rew_sensor_0
```

```
F rew_sensor_1
end

* Initial Probabilities  defined:
UP init_sensor_UP
F init_sensor_F
end


* Initial Probailities  assigned:
bind
  init_sensor_UP 0
  init_sensor_F 0
end

* Initial Probability: init
bind
  init_sensor_UP  1
  init_sensor_F  0
end


markov micro
UP F lam_mic
reward
UP rew_micro_0
F rew_micro_1
end
* Initial Probabilities  defined:
UP init_micro_UP
F init_micro_F
end

* Initial Probailities  assigned:
bind
  init_micro_UP 0
  init_micro_F 0
end

* Initial Probability: init
bind
  init_micro_UP  1
  init_micro_F  0
end

markov memory
UP F lam_mem
```

```
reward
UP rew_memory_0
F rew_memory_1
end
* Initial Probabilities  defined:
UP init_memory_UP
F init_memory_F
end

* Initial Probailities  assigned:
bind
  init_memory_UP 0
  init_memory_F 0
end

* Initial Probability: init
bind
  init_memory_UP  1
  init_memory_F  0
end

markov trans
UP F lam_tra
reward
UP rew_trans_0
F rew_trans_1
end
* Initial Probabilities  defined:
UP init_trans_UP
F init_trans_F
end

* Initial Probailities  assigned:
bind
  init_trans_UP 0
  init_trans_F 0
end


* Initial Probability: init
bind
  init_trans_UP  1
  init_trans_F  0
end

markov tinyos
G V lam_v
```

```
G FP lam_fp
V A lam_a
FP R r
FP F lam_f1
A D Cs*delta_s
A UC (1-Cs)*lam_uc
D G mu_s
R F (1-Cr)*mu_rj
R G Cr*mu_rj

reward
G rew_tinyos_0
V rew_tinyos_1
FP rew_tinyos_2
R rew_tinyos_3
A rew_tinyos_4
F rew_tinyos_5
UC rew_tinyos_6
D rew_tinyos_7
end

* Initial Probabilities  defined:
G init_tinyos_G
V init_tinyos_V
FP init_tinyos_FP
A init_tinyos_A
D init_tinyos_D
UC init_tinyos_UC
R init_tinyos_R
F init_tinyos_F
end

* Initial Probailities  assigned:
bind
  init_tinyos_G 0
  init_tinyos_V 0
  init_tinyos_FP 0
  init_tinyos_A 0
  init_tinyos_D 0
  init_tinyos_UC 0
  init_tinyos_R 0
  init_tinyos_F 0
end

* Initial Probability: init
bind
  init_tinyos_A  0
```

```
  init_tinyos_V  0
  init_tinyos_UC  0
  init_tinyos_FP  0
  init_tinyos_R  0
  init_tinyos_G  1
  init_tinyos_F  0
  init_tinyos_D  0
end

markov app
UP F lam_app
* Reward configuration  defined:
reward
UP rew_app_0
F rew_app_1
end
* Initial Probabilities  defined:
UP init_app_UP
F init_app_F
end

* Reward configuration  assigned:
bind
  rew_app_UP 0
  rew_app_F 1
end

* Initial Probailities  assigned:
bind
  init_app_UP 0
  init_app_F 0
end

* Initial Probability: init
bind
  init_app_UP  1
  init_app_F  0
end
bind
power 0
sensor 0
micro 0
memory 0
trans 0
tinyos 0
app 0
end
```

```
ftree singsen(t)
basic pow1 prob(power)
basic sen1 prob(sensor)
basic mic1 prob(micro)
basic mem1 prob(memory)
basic tra1 prob(trans)
basic tin1 prob(tinyos)
basic app1 prob(app)
or or3  tin1 app1
or or1  pow1 sen1  mic1  mem1 tra1
or or0  or1 or3
end

echo
**********************************************************
**********************
echo *********  Outputs  asked for the model:  singsen
**************
bind
rew_tinyos_0 0
rew_tinyos_1 0
rew_tinyos_2 0
rew_tinyos_3 0
rew_tinyos_4 0
rew_tinyos_5 1
rew_tinyos_6 1
rew_tinyos_7 0
rew_sensor_0 0
rew_sensor_1 1
rew_power_0 0
rew_power_1 1
rew_power_2 1
rew_power_3 0
rew_micro_0 0
rew_micro_1 1
rew_memory_0 0
rew_memory_1 1
rew_trans_0 0
rew_trans_1 1
rew_app_0 0
rew_app_1 1

end

func Reliability(t)  1-tvalue(t;singsen;t)
*loop t,1,2000,40
```

```
*expr Reliability(t)
*end

bind
p1 0
p 0
FTbat 0
FTpow 0
FTsen 0
FTmic 0
FTmem 0
FTtra 0
FTdet 0
FTfau 0
FTapp 0
FTrej 0
FT_Pow 0
FT_Tin 0
CFTbat 0
CFTpow 0
CFTsen 0
CFTmic 0
CFTmem 0
CFTtra 0
CFTdet 0
CFTfau 0
CFTapp 0
CFTrej 0

CFT_Pow 0
CFT_Tin 0

RRWsen 0
RAWsen 0
RRWbat 0
RAWbat 0
RRWpow 0
RAWpow 0
RRWmic 0
RAWmic 0
RRWmem 0
RAWmem 0
RRWtra 0
RAWtra 0
RRWdet 0
RAWdet 0
RRWfau 0
```

```
RAWfau 0
RRWrej 0
RAWrej 0
RRWapp 0
RAWapp 0
RRW_Pow 0
RAW_Pow 0
RRW_Tin 0
RAW_Tin 0

end

loop t,1,2000,40
*real computation
bind
  rew_power_0  assign_rew_power(1,-1,-1,0)
  rew_power_1  assign_rew_power(1,-1,-1,1)
  rew_power_2  assign_rew_power(1,-1,-1,2)
  rew_power_3  assign_rew_power(1,-1,-1,3)
  rew_sensor_0  assign_rew_sensor(2,-1,-1,0 )
  rew_sensor_1  assign_rew_sensor(2,-1,-1,1 )
  rew_micro_0  assign_rew_micro(3,-1,-1,0)
  rew_micro_1  assign_rew_micro(3,-1,-1,1)
  rew_memory_0  assign_rew_memory(4,-1,-1,0 )
  rew_memory_1  assign_rew_memory(4,-1,-1,1 )
  rew_trans_0  assign_rew_trans(5,-1,-1,0)
  rew_trans_1  assign_rew_trans(5,-1,-1,1)
  rew_tinyos_0  assign_rew_tynos(6,-1,-1,0)
  rew_tinyos_1  assign_rew_tynos(6,-1,-1,1)
  rew_tinyos_2  assign_rew_tynos(6,-1,-1,2)
  rew_tinyos_3  assign_rew_tynos(6,-1,-1,3)
  rew_tinyos_4  assign_rew_tynos(6,-1,-1,4)
  rew_tinyos_5  assign_rew_tynos(6,-1,-1,5)
  rew_tinyos_6  assign_rew_tynos(6,-1,-1,6)
  rew_tinyos_7  assign_rew_tynos(6,-1,-1,7)
  rew_app_0  assign_rew_app(7,-1,-1,0)
  rew_app_1  assign_rew_app(7,-1,-1,1)


battery  tvalue(t;power,BD)
powers  tvalue(t;power,F)
sensor  exrt(t;sensor)
micro  exrt(t;micro)
memory  exrt(t;memory)
trans  exrt(t;trans)
power  exrt(t;power)
tinyos  exrt(t;tinyos)
```

```
app  exrt(t;app)

battery1   tvalue(t;power,BD)
powers1  tvalue(t;power,F)
sensor1  exrt(t;sensor)
micro1  exrt(t;micro)
memory1  exrt(t;memory)
trans1  exrt(t;trans)
power1  exrt(t;power)
tinyos1  exrt(t;tinyos)
app1  exrt(t;app)
reliability Reliability(t)
unreliability 1-reliability

BIMpow  bimpt(t;singsen,pow1;t)
BIMsen  bimpt(t;singsen,sen1;t)
BIMmic  bimpt(t;singsen,mic1;t)
BIMmem  bimpt(t;singsen,mem1;t)
BIMtra  bimpt(t;singsen,tra1;t)
BIMapp  bimpt(t;singsen,app1;t)
BIMtin  bimpt(t;singsen,tin1;t)
CIFpow  cimpt(t;singsen,pow1;t)
CIFtin  cimpt(t;singsen,tin1;t)
end
*fine real computation -  Compute IM
*j:1-10 components of the  systems
*j=11 Power subsystem
*j=12 TinyOS subsystem
loop j,1,12,1
loop k,0,1,1
  bind
rew_power_0  assign_rew_power(1,k,j,0)
rew_power_1  assign_rew_power(1,k,j,1)
rew_power_2  assign_rew_power(1,k,j,2)
rew_power_3  assign_rew_power(1,k,j,3)
rew_sensor_0  assign_rew_sensor(2,k,j,0)
rew_sensor_1  assign_rew_sensor(2,k,j,1)
rew_micro_0   assign_rew_micro(3,k,j,0)
rew_micro_1   assign_rew_micro(3,k,j,1)
rew_memory_0  assign_rew_memory(4,k,j,0)
rew_memory_1  assign_rew_memory(4,k,j,1)
rew_trans_0  assign_rew_trans(5,k,j,0)
rew_trans_1  assign_rew_trans(5,k,j,1)
rew_tinyos_0  assign_rew_tynos(6,k,j,0)
rew_tinyos_1  assign_rew_tynos(6,k,j,1)
rew_tinyos_2  assign_rew_tynos(6,k,j,2)
rew_tinyos_3  assign_rew_tynos(6,k,j,3)
```

```
rew_tinyos_4  assign_rew_tynos(6,k,j,4)
rew_tinyos_5  assign_rew_tynos(6,k,j,5)
rew_tinyos_6  assign_rew_tynos(6,k,j,6)
rew_tinyos_7  assign_rew_tynos(6,k,j,7)
rew_app_0     assign_rew_app(7,k,j,0)
rew_app_1  assign_rew_app(7,k,j,1)

battery  tvalue(t;power,BD)
powers  tvalue(t;power,F)
sensor  exrt(t;sensor)
micro  exrt(t;micro)
memory  exrt(t;memory)
trans  exrt(t;trans)
power   wrapFT(exrt(t;power),j,k,11)
*power  exrt(t;power)
*tinyos  exrt(t;tinyos)
tinyos   wrapFT(exrt(t;tinyos),j,k,12)
app  exrt(t;app)
  p tvalue(t;singsen;t)
  p1 Store(k,p,p1)

  RRWbat  computeRRW(k,p,RRWbat,j,1,unreliability)
  RAWbat  computeRAW(k,p,RAWbat,j,1,unreliability)
  RRWpow  computeRRW(k,p,RRWpow,j,2,unreliability)
  RAWpow  computeRAW(k,p,RAWpow,j,2,unreliability)
  RRWsen  computeRRW(k,p,RRWsen,j,3,unreliability)
  RAWsen  computeRAW(k,p,RAWsen,j,3,unreliability)
  RRWmic  computeRRW(k,p,RRWmic,j,4,unreliability)
  RAWmic  computeRAW(k,p,RAWmic,j,4,unreliability)
  RRWmem  computeRRW(k,p,RRWmem,j,5,unreliability)
  RAWmem  computeRAW(k,p,RAWmem,j,5,unreliability)
  RRWtra  computeRRW(k,p,RRWtra,j,6,unreliability)
  RAWtra  computeRAW(k,p,RAWtra,j,6,unreliability)
  RRWdet  computeRRW(k,p,RRWdet,j,7,unreliability)
  RAWdet  computeRAW(k,p,RAWdet,j,7,unreliability)
  RRWfau  computeRRW(k,p,RRWfau,j,8,unreliability)
  RAWfau  computeRAW(k,p,RAWfau,j,8,unreliability)
  RRWrej  computeRRW(k,p,RRWrej,j,9,unreliability)
  RAWrej  computeRAW(k,p,RAWrej,j,9,unreliability)
  RRWapp  computeRRW(k,p,RRWapp,j,10,unreliability)
  RAWapp  computeRAW(k,p,RAWapp,j,10,unreliability)
  RRW_Pow  computeRRW(k,p,RRW_Pow,j,11,unreliability)
  RAW_Pow  computeRAW(k,p,RAW_Pow,j,11,unreliability)
  RRW_Tin  computeRRW(k,p,RRW_Tin,j,12,unreliability)
  RAW_Tin  computeRAW(k,p,RAW_Tin,j,12,unreliability)
```

```
  end
  expr p
  end
  bind
  FT p1-p
  FTbat   StoreFT(j,FT,FTbat,1)
  FTpow   StoreFT(j,FT,FTpow,2)
  FTsen  StoreFT(j,FT,FTsen,3)
  FTmic   StoreFT(j,FT,FTmic,4)
  FTmem   StoreFT(j,FT,FTmem,5)
  FTtra   StoreFT(j,FT,FTtra,6)
  FTdet  StoreFT(j,FT,FTdet,7)
  FTfau   StoreFT(j,FT,FTfau,8)
  FTrej   StoreFT(j,FT,FTrej,9)
  FTapp  StoreFT(j,FT,FTapp,10)
  FT_Pow  StoreFT(j,FT,FT_Pow,11)
  FT_Tin  StoreFT(j,FT,FT_Pow,12)

  CFTbat   computeCFT(FTbat*unreal_comp(lam_bd,t),unreliability)
  CFTpow   computeCFT(FTpow*unreal_comp(lam_pwr,t),unreliability)
  CFTsen  computeCFT(FTsen*sensor1,unreliability)
  CFTmic  computeCFT(FTmic*micro1,unreliability)
  CFTmem  computeCFT(FTmem*memory1,unreliability)
  CFTtra  computeCFT(FTtra*trans1,unreliability)
  CFTdet  computeCFT(FTdet*unreal_comp((1-Cs)*lam_uc,t),unreliability)
  CFTrej  computeCFT(FTrej*unreal_comp(r,t),unreliability)
  CFTfau   computeCFT(FTfau*unreal_comp(lam_f1,t),unreliability)
  CFTapp  computeCFT(FTapp*app1,unreliability)
******** Sottosistemi ********
CFT_Pow  computeCFT(FT_Pow*power1,unreliability)
 CFT_Tin  computeCFT(FT_Tin*tinyos1,unreliability)
 *CFT_Pow  (RAW_Pow-RRW_Pow)*power1
 *CFT_Tin  (RAW_Tin-RRW_Tin)*tinyos1
********* ----------  *********

  end
  end
bind

SIMpow simpt(singsen,pow1;power1)
SIMsen simpt(singsen,sen1;sensor1)
SIMmic simpt(singsen,mic1;micro1)
SIMmem simpt(singsen,mem1;memory1)
SIMtra simpt(singsen,tra1;trans1)
SIMtin simpt(singsen,tin1;tinyos1)
SIMapp simpt(singsen,app1;app1)
end
```

```
expr reliability
expr FTbat
expr FTpow
expr FTsen
expr FTmic
expr FTmem
expr FTtra
expr FTdet
expr FTfau
expr FTrej
expr FTapp
expr FT_Pow
expr FT_Tin
expr CFTbat
expr CFTpow
expr CFTsen
expr CFTmic
expr CFTmem
expr CFTtra
expr CFTdet
expr CFTfau
expr CFTrej
expr CFTapp
expr CFT_Pow
expr CFT_Tin
expr RRWbat
expr RRWpow
expr RRWsen
expr RRWmic
expr RRWmem
expr RRWtra
expr RRWdet
expr RRWfau
expr RRWrej
expr RRWapp
expr RAWpow
expr RAWbat
expr RAWsen
expr RAWmic
expr RAWmem
expr RAWtra
expr RAWdet
expr RAWfau
expr RAWrej
expr RAWapp
expr RRW_Pow
```

```
expr RAW_Pow
expr RRW_Tin
expr RAW_Tin
expr SIMpow
expr SIMsen
expr SIMmic
expr SIMmem
expr SIMtra
expr SIMtin
expr SIMapp
expr power1
expr sensor1
expr micro1
expr memory1
expr trans1
expr tinyos1
expr app1

end
end
```