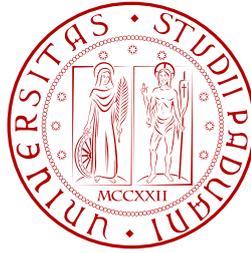


INTEGRAZIONE TRA SIMULAZIONE NUMERICA E
DATI REALI IN PROCEDURE AUTOMATICHE DI
COLLAUDO

RELATORE: Ch.mo Prof. Alessandro. Sona

LAUREANDO: Moreno Favaro

A.A. 2009-2010



UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI DI LAUREA

INTEGRAZIONE TRA SIMULAZIONE
NUMERICA E DATI REALI IN
PROCEDURE AUTOMATICHE DI
COLLAUDO

RELATORE: Ch.mo Prof. Alessandro. Sona

LAUREANDO: *Moreno Favaro*

Padova, 23 settembre 2010

Indice

Introduzione	V
L'azienda	V
L'idea	VI
1 Il simulatore NI Multisim	1
1.1 Introduzione a NI Multisim	1
1.2 Alcuni esempi	3
1.2.1 Amplificatore Invertente	3
1.2.2 Filtri e attenuazione del rumore	4
1.2.3 Modulatore e demodulatore	6
1.2.4 Contatore di impulsi a tre cifre	8
2 NI Labview	11
2.1 Introduzione a NI Labview	11
2.1.1 I Virtual Instruments	12
2.1.2 Le strutture	14
2.2 Multisim Connectivity Toolkit	18
2.2.1 I VI della toolkit	20
2.2.2 Schema a blocchi minimo	30
3 Conversione A→D e schede di acquisizione (ELVIS)	33
3.1 Conversione analogica-digitale	33
3.1.1 Il campionamento	33
3.1.2 La quantizzazione	35
3.2 Schede di acquisizione (ELVIS)	37

4	TASC	41
4.1	Circuito Multisim	41
4.2	Acquisizione e campionamento	43
4.3	Elaborazione dati	46
5	Misure	55
5.1	Ventola raffreddamento per CPU	55
5.2	Trasmissione segnale binario	59
	Conclusioni	66
	Bibliografia	67

Introduzione

L'azienda

IRS viene fondata a Padova nel 1993 da un gruppo di ingegneri con significative esperienze nei settori dell'informatica scientifica, delle misure, delle acquisizioni dati e nel controllo di processo. L'anno seguente diventa Alliance Member di National Instruments (NI), principale produttore al mondo di componenti e software per l'acquisizione dati. Nel 2006 diventa distributore nazionale dei prodotti NI Electronics Workbench per la progettazione di circuiti elettronici. Da gennaio 2009 National Instruments affida in esclusiva a IRS la rivendita dei propri prodotti hardware e software nel settore Accademic e Research. IRS organizza corsi di Labview e hardware National Instruments per studenti e professionisti. Le soluzioni che offre alla didattica sono Labview, Laboratorio per la misura e l'analisi di dati: NI ELVIS, NI Multisim, Utilboard e MeetVIEW. Ha collaborato con LEGO alla realizzazione di un kit di sviluppo su Robot comandato da LabVIEW. L'azienda fornisce soluzioni standard e sviluppate ad hoc per applicazioni didattiche e di ricerca per scuole, università e centri di ricerca offrendo servizi di consulenza e ingegneria nel campo del trattamento elettrostatico di polveri, gas di combustione e di inquinanti gassosi; in questo campo collabora con Università di Padova, CNR, ENEA, Ecole Supérieure d'Electricité, Stevens Institute of Technology e con le maggiori Aziende Elettriche d'Europa (Enel, EDF). Nel settore industriale fornisce diverse e collaudate soluzioni per il monitoraggio e la misura di grandezze fisiche. Produce applicazioni per collaudi, test qualità, test vita, monitoraggio e controllo di elettrodomestici, ma ha anche prodotto progetti per il monitoraggio dinamico delle strutture civili. IRS gestisce la progettazione dalla fase di analisi/studio/progetto alla fase di realizzazione ed installazione sul campo.

L'idea

Il Product Manager EWB (*linea di prodotti NI Multisim e Ultiboard*) Massimo Rapini è l'ingegnere IRS che ha pensato alla possibilità di integrare le capacità di progettazione e simulazione di Multisim con l'analisi e l'elaborazione dati che offrono Labview e le schede di acquisizione. L'obbiettivo che si vuole raggiungere è quello di poter collaudare e monitorare un circuito simulato con dati reali acquisiti usando Labview. Questo tipo di approccio all'ingegnerizzazione permette di considerare, già in fase di simulazione, tutti gli aspetti imprevisi che possono essere presenti in segnali reali acquisiti dall'ambiente e non con semplici segnali ideali. In fase di progettazione di un circuito elettronico, spesso, si considerano solo delle stime del rumore e delle interferenze che possono essere presenti nell'ambiente dell'installazione, ma collaudare un circuito, già in fase di sviluppo, con il vero segnale al quale sarà sottoposto, o può far risparmiare tempo nell'analisi e nello studio del circuito. Uno strumento che consente la taratura di un circuito in determinate condizioni o con determinati segnali può essere utile in progettazione, ma può essere utilizzato anche nella didattica scolastica per lo studio di classici schemi elettrici come l'amplificazione, il filtraggio, la modulazione e la demodulazione. Nel capitolo 5 verranno illustrati alcuni esempi in cui l'integrazione tra simulazione numerica e dati reali ha portato un vantaggio nello studio, nella progettazione e nel collaudo di circuiti elettronici.

Capitolo 1

Il simulatore NI Multisim

1.1 Introduzione a NI Multisim

SPICE (*Simulation Program with Integrated Circuit Emphasis*) è un programma di simulazione circuitale sviluppato nel 1975 e oggi distribuito sotto licenza BSD. Nel corso degli anni è divenuto lo strumento standard per la simulazione di circuiti elettronici analogici. La simulazione circuitale si occupa dello studio di circuiti elettrici per mezzo di modelli matematici. È basato sui metodi di analisi standard della teoria dei circuiti. La simulazione è uno strumento indispensabile per la progettazione di prototipi soprattutto di medie o grandi dimensioni. La capacità di poter emulare il comportamento di un circuito elettrico reale in condizioni ideali rende facile e veloce la prototipizzazione del progetto. NI Multisim è un software creato per la progettazione di circuiti elettronici che consente esperimenti ed analisi del funzionamento circuitale grazie a strumenti virtuali. NI Multisim aiuta il progettista a identificare errori prima che il prodotto venga immesso nel mercato e a ridurre il time-to-market velocizzando le fasi di ingegnerizzazione e permettendo di predire, e meglio comprendere, il comportamento del circuito. In molti istituti di scuola media superiore e università viene utilizzato NI Multisim per dare più incisività e spessore alle lezioni. NI Multisim dispone di un ampio set di simboli interattivi, animati, virtuali, a rottura e 3D, per rendere la simulazione sempre più reale e completa, infatti, ad esempio, è possibile utilizzare display a 7 segmenti, led, lampade, ecc. che si accendono realmente sullo schermo durante la simulazione. Analogamente sono disponibili interruttori, pulsanti, potenziometri, condensatori e induttori variabili che è possibile controllare

1. IL SIMULATORE NI MULTISIM

direttamente da tastiera durante la simulazione per agire sul circuito. I componenti virtuali sono componenti dotati di valori elettronici ideali che permettono di simulare il comportamento del circuito in condizioni ideali senza influenza di tolleranze, valori parassiti, ecc. Nelle librerie di NI Multisim sono disponibili anche componenti reali a rottura che si rompono o bruciano (come accade nella realtà) per valori di corrente/tensione non tollerati. NI Multisim consente la creazione di sottocircuiti o scatole-nera contenenti una parte del circuito. Questa caratteristica può essere utilizzata per semplificare la comprensione di un circuito mediante la realizzazione di una struttura a blocchi. L'interfaccia di NI Multisim è mostrata in figura 1.1.

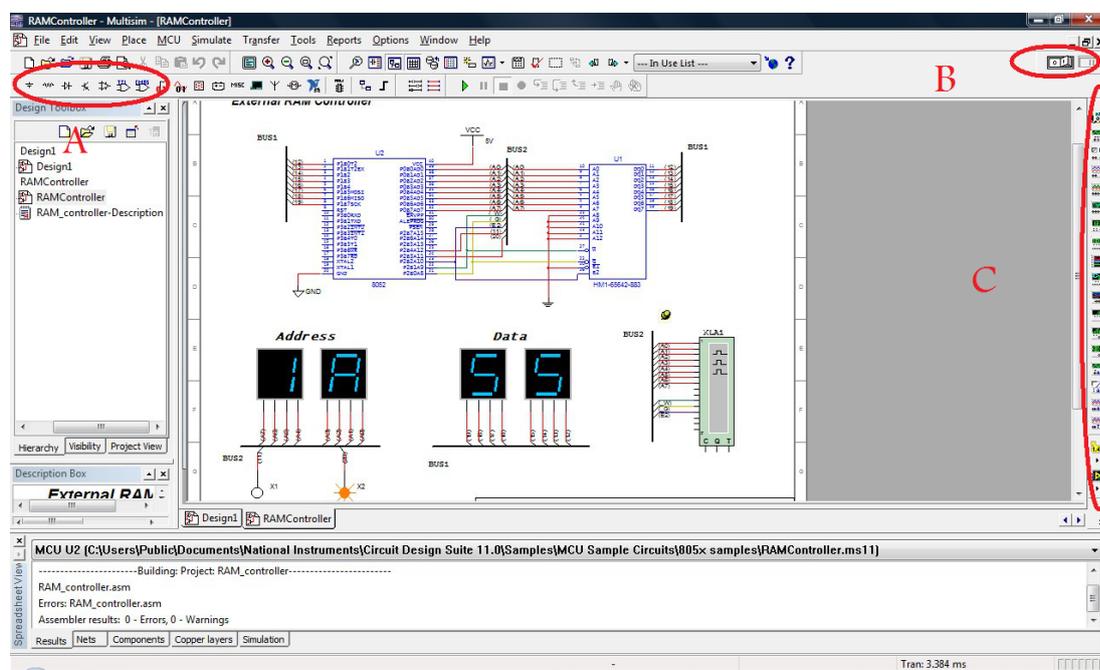


Figura 1.1: Interfaccia grafica di NI Multisim

Oltre alla parte centrale di figura 1.1, dove è stato creato un circuito digitale, sono state evidenziate in rosso tre sezioni importanti:

- A) la toolbar serve per aprire il browser dei componenti presenti nel database e selezionare in base alla categoria e alla famiglia il componente desiderato;
- B) nella sezione indicata viene avviata, arrestata o messa in pausa la simulazione¹;
- C) la terza sezione mostra la suite degli strumenti utilizzabili dal simulatore.

¹Se la simulazione è avviata non è possibile cambiare i componenti o i valori dei componenti

NI Multisim mette a disposizione dell'utente un set completo di 18 strumenti di simulazione (oscilloscopio, multimetro, analizzatore di spettro, ecc.) per consentire lo studio completo (spesso non possibile con gli strumenti reali disponibili in laboratorio) di qualsiasi tipo di circuito. Grazie a tale caratteristica di NI Multisim è possibile ridurre enormemente i costi per l'acquisto di costosi strumenti di laboratorio.

1.2 Alcuni esempi

In questa sezione del capitolo verranno illustrati e studiati alcuni semplici circuiti simulati con NI Multisim utili allo studio delle potenzialità e dei limiti dell'integrazione con Labview e le schede di acquisizione.

1.2.1 Amplificatore Invertente

Il circuito di figura 1.2 mostra il comportamento di un circuito con un amplificatore operazionale in configurazione invertente con guadagno di 5 e sottoposto ad un segnale d'ingresso (mostrato in blu) sinusoidale di frequenza di 1kHz a 2 V_{pp} . All'uscita dell'amplificatore è collegata l'entrata dell'oscilloscopio di NI Multisim che mostra un segnale in rosso di 10 V_{pp} sfasato di 180° rispetto all'ingresso. NI Multisim offre uno strumento (Tools→Circuit Wizard) in grado di costruire lo schema di amplificazione che rispetti le specifiche di richiesta. Basta inserire la configurazione dell'amplificatore, la frequenza del segnale d'ingresso e il guadagno ideale dell'amplificatore. Per ottenere un segnale d'uscita più grande è necessario arrestare la simulazione, cambiare il valore di una delle 2 resistenze sapendo che $\frac{V_o}{V_i} = \frac{-R_2}{R_1}$ e far ripartire la simulazione. Questo tipo di operazione può essere ripetuta fino ad ottenere il segnale d'uscita desiderato. Data la semplicità del circuito non sembra necessario ripetere più volte l'operazione di taratura dei parametri, ma la possibilità di fare diversi tentativi, senza modificare un circuito reale, rende la simulazione uno strumento estremamente prezioso. In figura 1.2 si può notare in basso a destra una piccola scritta: Trans:2.284ms. Questa cifra indica numero e unità di misura del tempo trascorso dall'inizio della simulazione. La simulazione SPICE comporta grosso dispendio di risorse di CPU e RAM in fase di simulazione, implicando che il tempo necessario per eseguire tutti i complessi calcoli differenziali, sia maggiore del tempo trascorso realmente. Questo per

1. IL SIMULATORE NI MULTISIM

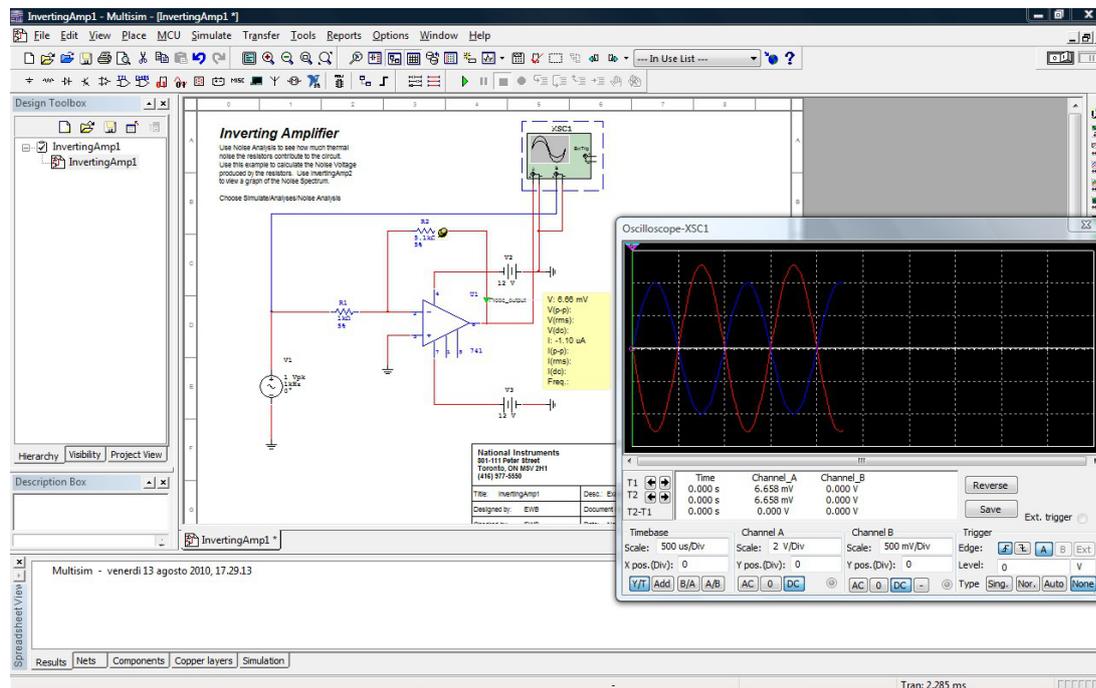


Figura 1.2: Schema di amplificazione invertente

indicare il fatto che la simulazione non è in tempo reale ma che è possibile settare il sincronismo di simulazione usando le *initial condition*, il *maximum timestep* e l'*initial time step* dal menu *Simulated*. Queste indicazioni temporali saranno da considerare quando la simulazione verrà gestita da Labview. Di default i generatori di tensione e corrente sono dei generatori ideali, il che vuol dire che nel circuito non vengono presi in considerazione rumori e disturbi. Le probes (sonde) sono degli strumenti, disegnati come delle piccole frecce verdi, che hanno lo scopo di far leggere all'utente i valori istantanei di tensione, corrente e frequenza del segnale.

1.2.2 Filtri e attenuazione del rumore

Uno dei problemi più frequenti nell'elettronica analogica e digitale è la sovrapposizione di segnali rumorosi al segnale utile che si vuole utilizzare. La soluzione più economica e veloce è l'utilizzo di un filtro che attenui l'ampiezza della parte rumorosa del segnale lasciando il più possibile inalterata la componente spettrale del segnale utile. NI Multisim Wizard offre all'utente un grosso aiuto nella progettazione di filtri analogici. È possibile scegliere il tipo di filtro, le frequenze

di taglio e il valore di attenuazione delle componenti fuori e dentro la banda. I generatori di tensione e corrente messi sul circuito sono di default ideali, ma cliccando col tasto destro sul generatore e scegliendo proprietà, è possibile inserire dei parametri che simulano l'effetto del rumore e della distorsione. Questi parametri simulano soltanto un rumore. NI Multisim Wizard è in grado di progettare filtri passivi RLC fino a un numero massimo di 10 componenti LC. Il grado del filtro è pari alla somma dei componenti LC, più alto è il grado del filtro, più componenti sono necessari per realizzare il filtro, maggiore è la pendenza del modulo del filtro nel dominio delle frequenze.

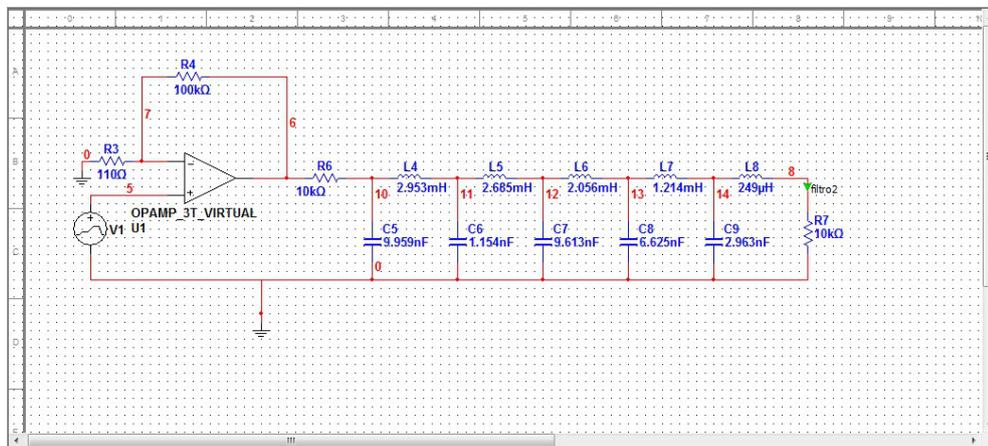


Figura 1.3: Filtro elimina rumore

Questo circuito ha un amplificatore operazionale in ingresso, successivamente il segnale entra nel vero e proprio filtro passivo. NI Multisim offre un altro strumento molto utile nell'analisi circuitale: l'AC Sweep (Simulate→Analysis→AC Analysis). Osservando la figura 1.4 si può notare che il modulo del segnale d'ingresso viene amplificato se la frequenza del segnale è minore della banda del filtro. Questo tipo di filtro è molto comune perché solitamente il rumore è un segnale che ha componenti spettrali in tutte le frequenze. In genere i filtri attenuano leggermente anche all'interno della banda passante, per questo il segnale viene prima amplificato e poi filtrato, tanto da compensare l'attenuazione indesiderata del filtro all'interno della banda passante. Il circuito in esame permette di attenuare le componenti ad alta frequenza e amplificare la parte del segnale a bassa frequenza eliminando una buona porzione del rumore. Questo circuito (filtro+amplificatore) verrà utilizzato in un caso pratico per eliminare del rumore

1. IL SIMULATORE NI MULTISIM

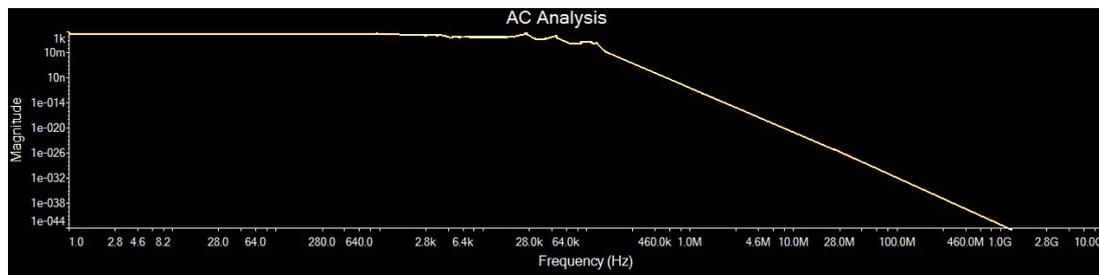


Figura 1.4: AC Analysis del modulo del filtro

ed avere un segnale pulito e amplificato. Anche i valori dei singoli componenti verranno discussi e tarati nel paragrafo 5.1.

1.2.3 Modulatore e demodulatore

Nella sezione Help→Find Examples di NI Multisim è disponibile un circuito interessante e molto usato che esegue la modulazione AM di un segnale analogico o digitale. La modulazione AM è uno dei sistemi utilizzati per trasmettere informazioni utilizzando radiofrequenze. Uno schema di modulazione AM è mostrato in figura 1.5. La logica della modulazione AM prevede di moltiplicare il segnale

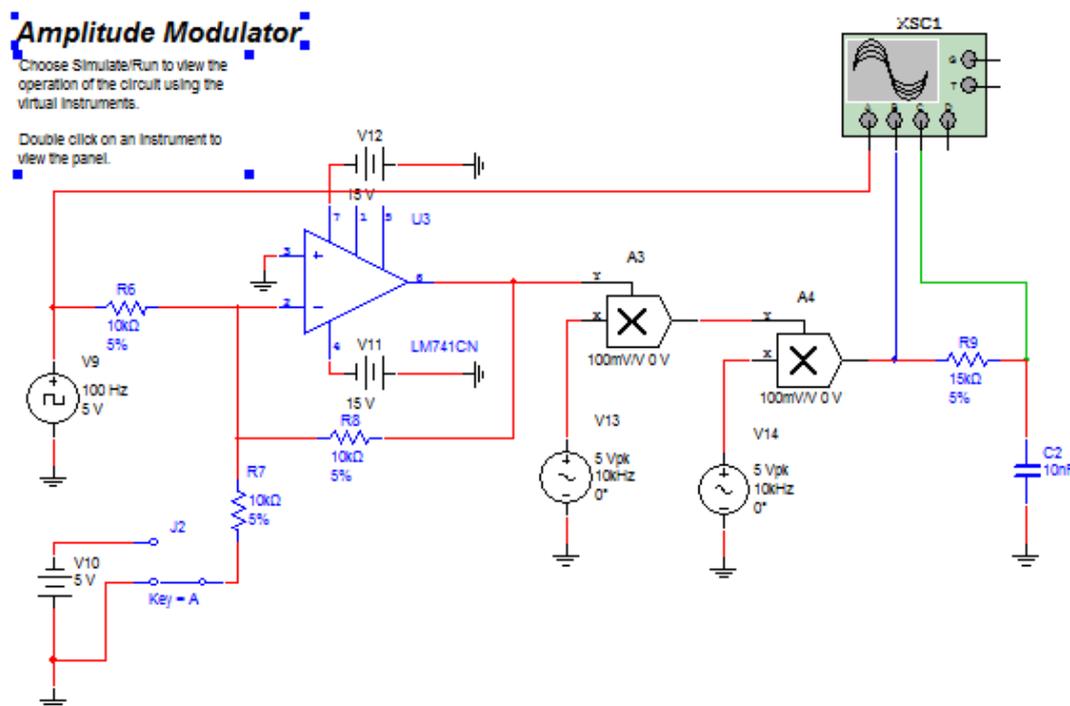


Figura 1.5: Schema di modulazione e demodulazione AM

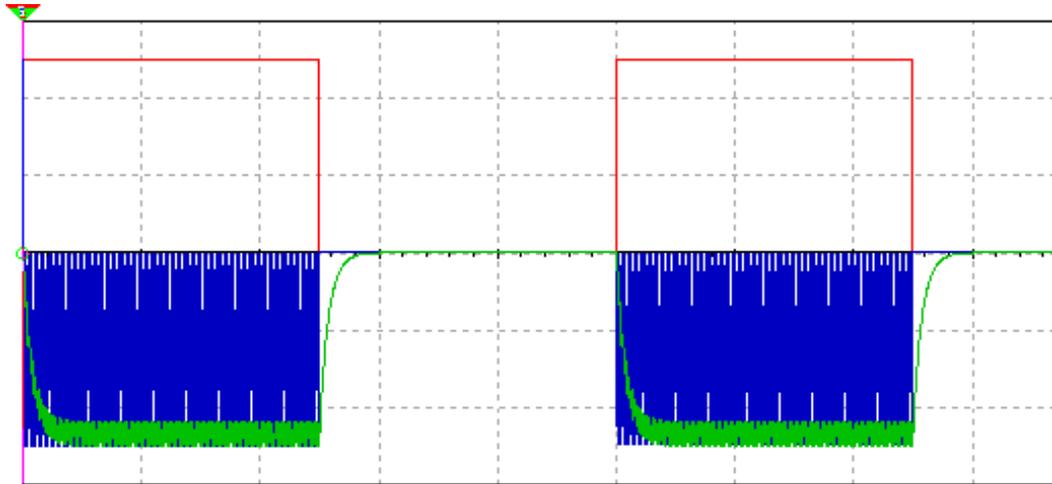


Figura 1.6: Segnale sorgente (rosso), segnale modulato (blu) e segnale demodulato (verde)

di interesse per un segnale ad alta frequenza chiamato portante. Nell'esempio di figura 1.5 il segnale digitale viene amplificato da un amplificatore operazionale invertente di guadagno unitario. Successivamente il segnale viene moltiplicato per un segnale sinusoidale a frequenza di 10kHz. Il segnale risultante sarà nullo nei punti in cui il segnale digitale è a livello logico basso e sarà una sinusoide di ampiezza 5V per l'ampiezza del valore logico alto. Il moltiplicatore di segnali divide l'uscita di un fattore 10. Questa operazione viene fatta una seconda volta ottenendo un segnale a frequenza di 100MHz non nullo solo negli istanti in cui il segnale digitale è non nullo. L'uscita dei moltiplicatori è filtrata dal circuito RC che attenua anche l'ampiezza del segnale modulato anche all'interno della banda passante, come mostrato in figura 1.6. Il segnale sorgente digitale è mostrato in figura in rosso, il segnale all'uscita dei moltiplicatori è mostrata in blu mentre il segnale in uscita dal filtro è il segnale verde il figura 1.6 Dal segnale verde sarà semplice ritornare al segnale originale con un semplice comparatore con soglia a zero. La modulazione AM è molto utile nella trasmissione dati, ma per la realizzazione di questo progetto non si possono trascurare gli effetti del rumore. Spesso un segnale modulato e trasmesso non ha ampiezze elevate e serve un filtro per eliminare il rumore al di fuori della banda del segnale utile. Questo circuito è proposto negli esempi di Multisim per vedere la simulazione, un foglio elettronico lo strumento oscilloscopio di NI Multisim, ma servirà solo come punto di partenza per il progetto nella sezione 5.2 per intuire come funziona la modulazione AM.

1.2.4 Contatore di impulsi a tre cifre

Lo schema di figura 1.7 è un circuito simulato che è stato realizzato anche fisicamente. Il segnale periodico d'ingresso entra direttamente nel piedino di *clock*, mentre i pin di *cten*, *U-D* e *Load* servono a resettare, incrementare o decrementare il conteggio e a metterlo in pausa. Nella figura 1.7 i display indicano un numero uguale a 109. Il segnale d'ingresso è un'onda quadra a frequenza esattamente di 1kHz. La prova che il circuito funziona correttamente è che il valore assunto da *Tran* è esattamente uguale a 0.109 s. 109 è il numero esatto di periodi di un segnale a frequenza di 1kHz che dura 0.109 secondi. Lo scopo da raggiungere è costruire uno strumento reale che possa fare delle misure reali su una grandezza reale, cioè considerando tolleranze, rumori, disturbi ecc e confrontarlo poi con il circuito simulato e con strumenti di precisione. Il circuito è stato progettato per essere usato come frequenzimetro. In realtà è un contatore di impulsi, ma se l'intera simulazione dura esattamente un secondo, se si è certi che il segnale d'ingresso sia periodico e di frequenza minore o uguale di 1kHz (con 3 display ci sono esattamente 1000 combinazioni), allora l'indicazione data dai display a 7 segmenti sarà la misura di frequenza del segnale. L'errore massimo sarà pari al peso della cifra meno significativa che è di 1 Hz.

Lo schema di figura 1.7 è stato realizzato realmente ed utilizzato come frequenzimetro inserendo in ingresso un segnale reale della durata esatta di un secondo. Come verrà discusso nell'ultimo capitolo, questo circuito funzionerà correttamente come frequenzimetro ma non per tutti i tipi di segnali. In questo paragrafo vengono solo introdotte la sua logica e il suo schema. Il funzionamento di questo circuito è dovuto alla logica degli integrati 74LS190 che ad ogni impulso di clock generano una combinazione di tensioni nei 4 pin di uscita collegati al display a sette segmenti, il quale aumenta il numero visualizzato di 1 ad ogni impulso di clock. L'integrato genera nel pin RCO un impulso quando il numero visualizzato nel display passa da 9 a 0. Questo impulso diventa il clock del 74LS190 seguente che farà in modo di visualizzare sul relativo display un numero che incrementa ogni 10 clock del primo. Questo vuol dire che il secondo integrato conteggia le decine di clock. Allo stesso modo il terzo integrato conterà le centinaia e via via. Nell'esempio preso in considerazione si usano solo tre integrati con i relativi tre display per motivi che saranno noti nel paragrafo 5.1.

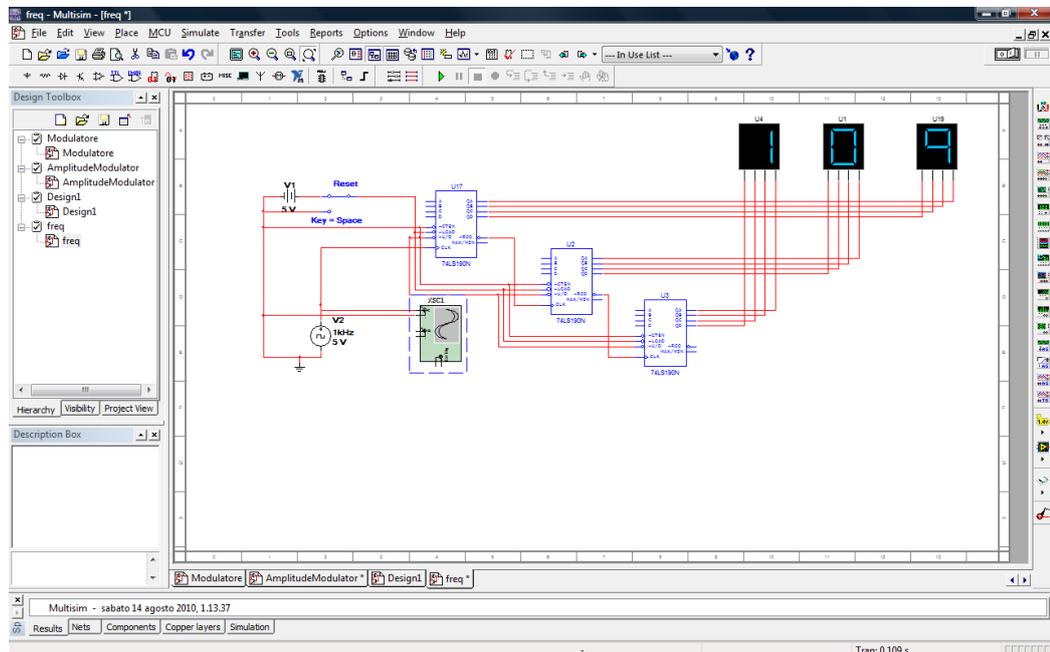


Figura 1.7: Contatore di impulsi a 3 cifre

Capitolo 2

NI Labview

2.1 Introduzione a NI Labview

LabVIEW (abbreviazione di Laboratory Virtual Instrumentation Engineering Workbench) è un ambiente di programmazione grafica che include collegamenti e icone grafiche per lo sviluppo di un'ampia gamma di sistemi avanzati di controllo, misura e test. Tale linguaggio grafico viene chiamato Linguaggio G. Un programma o sottoprogramma G, denominato VI (Virtual Instrument), non esiste sotto forma di testo, ma può essere salvato solo come un file binario, visualizzabile e compilabile solo da LabVIEW. LabVIEW è compatibile con migliaia di dispositivi hardware e include librerie per analisi avanzate e per la visualizzazione dei dati. La piattaforma di LabVIEW è scalabile su target multipli e sistemi operativi ed è uno standard per la programmazione grafica fin dal 1986. La definizione di strutture dati ed algoritmi avviene con icone e altri oggetti grafici, ognuno dei quali incapsula funzioni diverse, uniti da linee di collegamento (wire), in modo da formare una sorta di diagramma di flusso. Tale linguaggio viene definito dataflow (flusso di dati), in quanto la sequenza di esecuzione è definita e rappresentata dal flusso dei dati stessi attraverso i fili monodirezionali che collegano i blocchi funzionali. Poiché i dati possono anche scorrere in parallelo attraverso blocchi e fili non consecutivi, il linguaggio realizza spontaneamente il multithreading senza bisogno di esplicita gestione da parte del programmatore. La semplicità di programmazione (abbastanza intuitiva in quanto modellata su un diagramma di flusso), la semplicità di utilizzo (l'utente dispone di uno strumento virtuale disegnato sullo schermo del computer) e la grande versatilità, hanno reso LabVIEW

molto impiegato e diffuso nell'ambito dell'acquisizione dati e nel controllo dei processi industriali. È determinante anche la velocità di realizzazione del software, che si limita spesso al semplice collegamento di blocchi già pronti, grazie alla vasta libreria di funzioni predefinite e driver per la gestione dell'hardware. LabVIEW è completamente integrato per la comunicazione con l'hardware di tipo GPIB (488.2), VXI, PXI, RS-232, RS-485 e dispositivi DAQ plug-in.

2.1.1 I Virtual Instruments

I Virtual Instruments, o più brevemente VI, sono degli strumenti virtuali che permettono test analisi e misure di dati reali. Tipicamente sono strutturati in due parti: *Pannello frontale* e *Diagrammi a blocchi*. Il primo è composto dai controlli, comandi e dalla visualizzazione dei risultati comandabili e osservabili dall'utilizzatore dello strumento, mentre il secondo è lo schema dei collegamenti logici e matematici degli elementi che portano alla visualizzazione dei risultati. Lo schema a blocchi permette l'implementazione di algoritmi anche molto complessi utilizzando i VI installati nelle librerie che appaiono cliccando col tasto destro del mouse sullo schema a blocchi. Questi blocchi devono essere collegati tra loro con i *wire*. Ogni VI non viene eseguito fino a che non sono presenti tutti i parametri necessari. I tipi di dati che Labview riconosce sono numerici senza segno, numeri con virgola mobile, stringhe e booleani. Alcuni VI prevedono la generazione di errori. Gli errori sono un tipo di dati che raggruppano un booleano (errore sì o no), un numero senza segno (codice errore) e una stringa (messaggio di errore). La generazione di errori può dipendere da un errore prevedibile, e quindi gestibile, dal programmatore, come un valore sbagliato inserito dall'utente, oppure un errore non gestibile come la rottura di un dispositivo connesso a labview.

Labview offre *Context Help* agli sviluppatori. Premendo Ctrl+H su un qualsiasi elemento presente sullo schema a blocchi o sul pannello frontale, si apre una piccola finestra che definisce e descrive l'elemento selezionato. Spostando il cursore del mouse sopra altri elementi, *Context Help* aggiorna il riferimento e fornisce le veloci istruzioni di base del nuovo elemento al di sotto del cursore. Questa strategia permette al programmatore di vedere quali parametri sono fondamentali e una breve spiegazione sullo strumento selezionato. *Context Help* offre anche un link diretto a *Detailed Help* che è l'help di Labview che definisce dettagliatamente tutti i VI delle librerie fornendo informazioni dettagliate sulla struttura,

parametri minimi di cui necessita il VI e le funzionalità. In entrambi gli elemen-

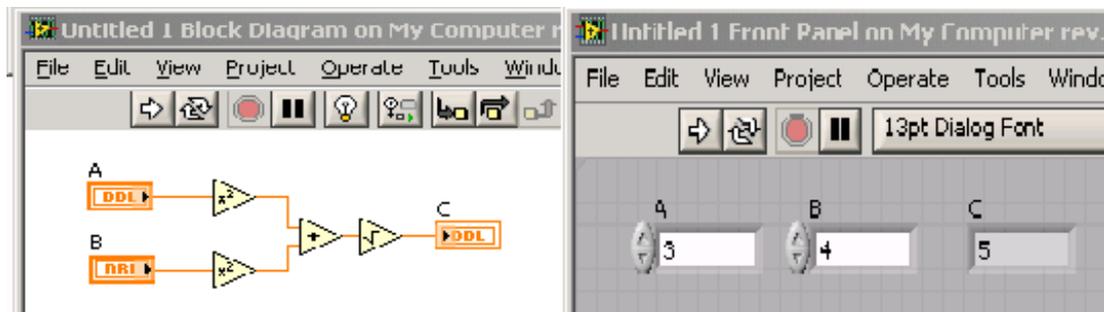


Figura 2.1: A sinistra lo schema a blocchi (algoritmo); a destra il pannello frontale (dati e risultati)

ti sono presenti dei pulsanti importanti per l'esecuzione di un VI. Questi sono: *Run*, rappresentato da una freccia bianca che punta verso destra ed ha la funzione di eseguire il VI; *Run Continuously*, rappresentato da una freccia bianca ciclica e tonda che ha la funzione di avviare il VI ripetutamente fino a che non viene ripremuto il pulsante; *Abort Execution*, rappresentato da un pallino rosso che ferma l'esecuzione istantaneamente; *Pause*, che mette in pausa la simulazione e la ripristina al secondo click.

Nel diagramma a blocchi sono presenti altri pulsanti molto utili nell'individuazione di errori logici, come: *HighLight Execution*, rappresentato da una piccola lampadina che avvia l'esecuzione del VI rallentata mostrando al programmatore il percorso del flusso di dati istante per istante, rappresentandolo con dei pallini che scorrono lungo i collegamenti; *Start Single Step*, che avvia l'esecuzione del VI passo passo mostrando al programmatore l'evoluzione logica della struttura creata. Questi pulsanti sono molto utili per individuare errori logici.

Un VI può essere salvato e utilizzato come *SubVI* da un VI più grande che lo contiene. Questo utilizzo è mostrato in figura 2.2 e mostra quanto la struttura sia molto più leggera da leggere e quindi comoda se si cerca un errore. All'interno di SubVI possono essere presenti altri SubVI creando una struttura ad albero che si può vedere cliccando su *View* → *VI Hierarchy*. I SubVI permettono al programmatore di usare strutture create in precedenza e utilizzarle in nuovi programmi velocizzando e alleggerendo la costruzione del VI. Quando si definisce un SubVI è necessario definire quali siano gli ingressi (obbligatori o meno), quali siano le uscite. Inoltre è possibile utilizzare una casella di testo, utile ai programmatori

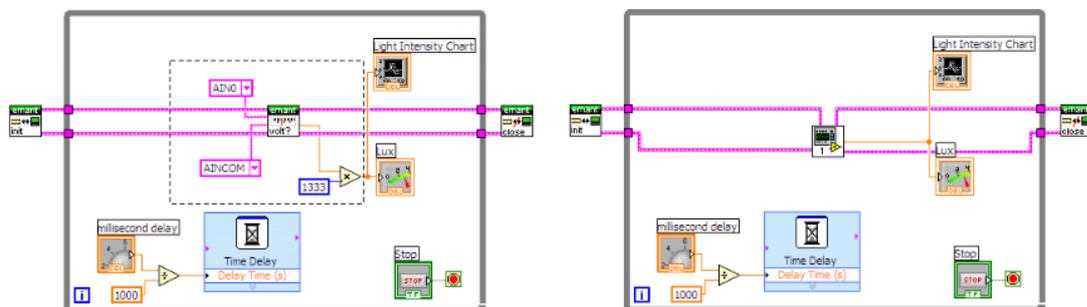


Figura 2.2: Creazione di un SubVI

che utilizzeranno il SubVI, che ne descrive funzionamento e logica. L'esempio di figura 2.3 mostra un semplice algoritmo che elabora dei dati casuali attraverso funzioni matematiche intuitive e ne mostra sul monitor i risultati. L'algoritmo di questo VI prevede la generazione casuale di un numero, compreso tra 0 e 1, che viene moltiplicato per 100 e di cui viene calcolato il valore intero, che viene convertito in string, concatenato con My lucky number is... e visualizzato. Il valore numerico viene poi confrontato con il numero 5 che è un valore deciso dall'utente. L'algoritmo accende la luce verde se il numero elaborato è maggiore di quello scelto dall'utente. In questo esempio vengono utilizzati dati (numeri, stringhe e valori booleani), indicatori (il grafico, l'indicatore di stringa e la luce verde) e controlli, cioè dati gestibili dall'utente (il valore 5).

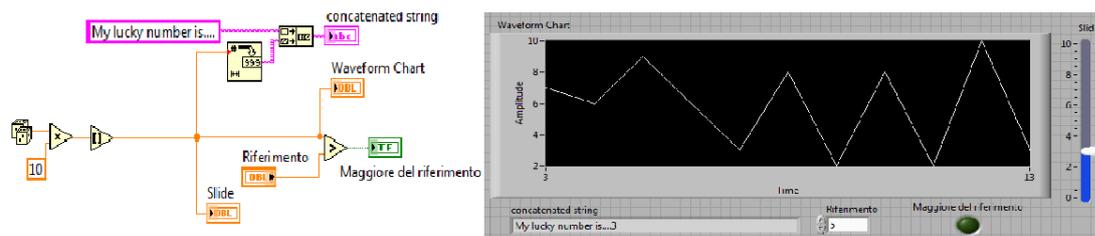


Figura 2.3: Esempio di schema logico a blocchi e pannello frontale con indicatori

2.1.2 Le strutture

Labview è un linguaggio di programmazione grafica che implementa molte delle strutture tipiche di linguaggi più comuni come Java e C. Labview ha i più classici cicli *For* e *While* o strutture decisionali tipo *Case* o *Timed Structures* o *Events Structures* oppure *Flat Sequence*. Queste strutture permettono l'evoluzione logica

dell'algoritmo. In seguito ne verranno ricordate alcune¹.

Ciclo FOR: si usa quando si deve rispettare una sequenza di operazioni per un numero N di volte, conosciuto dall'utente o generato da elaborazioni precedenti del VI. Si accede a questa struttura dal percorso All Functions → Structures → For loop e la si dispone sul diagramma a blocchi trascinandola. Nel punto indicato da N, si collega una costante (o il risultato di qualsiasi elaborazione che restituisce un numero) che rappresenta il numero di iterazioni che il ciclo eseguirà. Il quadratino in basso a sinistra invece restituisce l'indicazione sull'iterazione corrente. Tutto il codice inserito all'interno sarà eseguito quindi N volte. Il passaggio di valori all'interno del ciclo avviene semplicemente collegando due punti come si è fatto sinora. Il passaggio di valori all'esterno è invece più complesso. È possibile scegliere di trasferire solo un valore generato nell'ultima iterazione o tutta la raccolta di valori generati nel corso delle N iterazioni. Nell'immagine di figura 2.4 vengono generati 10 valori casuali, uno ogni secondo. In figura 2.4, se il quadratino è pieno vuol dire che verrà passato all'esterno solo l'ultimo valore, se il quadratino contiene all'interno le parentesi quadre vuol dire che passerà un array i cui elementi saranno i valori casuali generati.

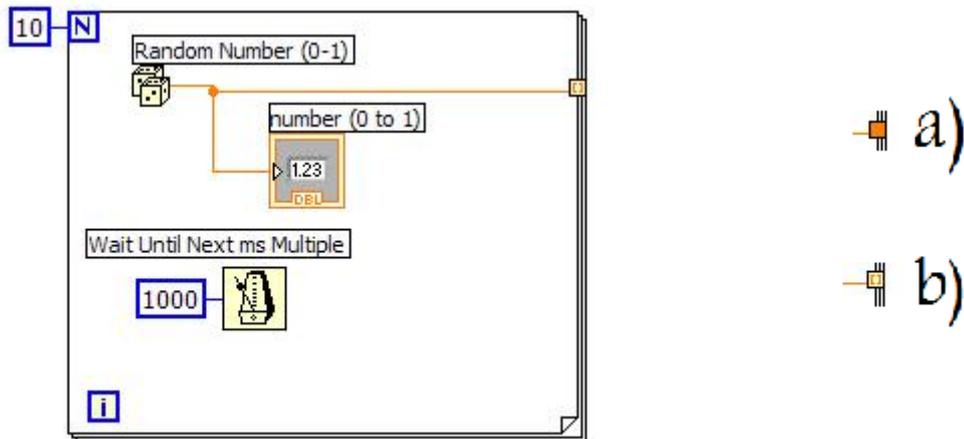


Figura 2.4: Esempio di ciclo For

¹Le strutture Case, For e While sono le principali di ogni linguaggio di programmazione. Labview offre diverse strutture dati, ma verranno descritte solo le principali per fare intuire al lettore la logica di TASC nel capitolo 4

Ciclo WHILE: come per il Ciclo For, tutto quello che si trova all'interno del riquadro while verrà ripetuto ma questa volta finchè non si avvera una condizione booleana. Si procede subito con un esempio: si accede a questa struttura dal percorso All Functions → Structures → While loop e la si dispone sul diagramma a blocchi trascinandola. Labview collegherà da solo un controllo sul valore booleano che determina la fine del ciclo. Il ciclo viene eseguito una volta al secondo sino a quando non verrà premuto il pulsante di STOP. Il passaggio di valori all'interno del ciclo funziona esattamente come il Ciclo For. Anche nel passaggio dei valori in uscita si possono scegliere le stesse opzioni per il Ciclo For con in più la modalità *Shift Register*, come mostrato in figura 2.5. Questa modalità permette di memorizzare l'ultimo valore di uscita e riportarlo in ingresso. Nell'esempio di figura i dati in uscita saranno rispettivamente numeric 2 e numeric (00;10;21;32;43;...). Questo ciclo permette di strutturare l'esecuzione del VI come macchina a stati utilizzata in TASC, che verrà spiegata tra breve.

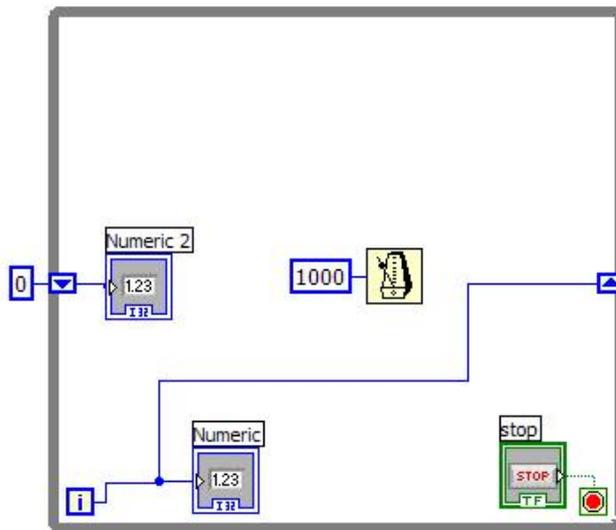


Figura 2.5: Esempio di utilizzo del ciclo While

Struttura CASE: la struttura Case permette l'elaborazione condizionata dei sub-diagrammi a seconda del valore assunto da una variabile di controllo. La variabile di controllo può essere un numero intero, un booleano, una stringa, un dato di tipo enumerato o un errore. In numero di sub-diagrammi dipende dal tipo di variabile di controllo e un solo sub-diagramma alla volta è visibile. Se la variabile è di tipo booleano ci possono essere solo due sub-diagrammi. In questo

caso la struttura Case è identica alla struttura *if... then... else..* di linguaggi di programmazione testuale.

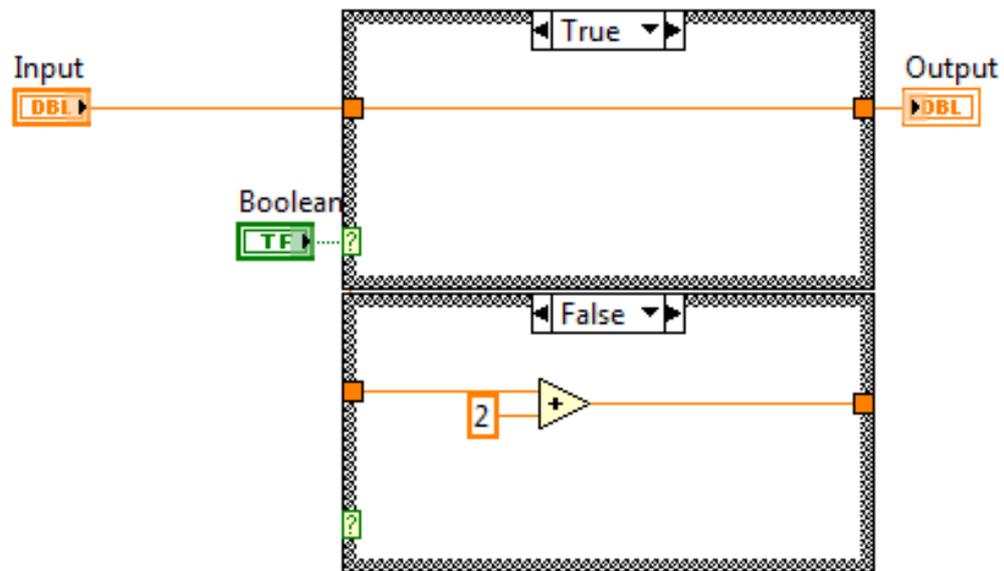


Figura 2.6: Esempio di struttura Case

Macchine a stati: la macchina a stati è una struttura logica che sfrutta il ciclo While e la struttura Case e che permette una programmazione leggera e facilmente leggibile da qualsiasi programmatore. In figura 2.7 è mostrato l'utilizzo dei shift register come pilota degli stati della macchina. Il valore 0 a sinistra è il caso di default da dove parte la macchina. La struttura case entra in zero ed esegue le istruzioni del sub-diagramma di 0 dove viene anche deciso lo stato successivo che viene indicato dallo shift register di destra. A questo punto il primo ciclo While termina, lo shift register di destra copia il suo valore sullo shift di sinistra e comincia il secondo ciclo While. La struttura Case legge il nuovo valore numerico (che è quello deciso nel caso 0), entra in quel caso, svolge le istruzioni di quel sub-diagramma dove viene deciso il valore che dovrà essere disponibile alla struttura Case nel ciclo successivo. Questo avviene ininterrottamente fino a quando non viene premuto il pulsante di stop. Nell'esempio di figura 2.7 la macchina parte dallo stato 0 in cui viene solo settata a null una stringa indicizzata a 1 lo shift register. Termina il primo ciclo e comincia il secondo. A questo punto lo shift register è di valore uguale a 1 e la struttura Case entra nel case

1. In questo Case è stata inserita una struttura *Event* che si comporta come una Case ma la decisione sul caso da eseguire non viene decisa dalla variabile di controllo, ma da un evento come la pressione di un tasto, il cambiamento di un valore numerico e l'arrivo di un errore. In questi casi gestiti il software entrerà nel case specifico pensato dal programmatore e farà in modo che la macchina si comporti come previsto potendo gestire anche eventuali errori.

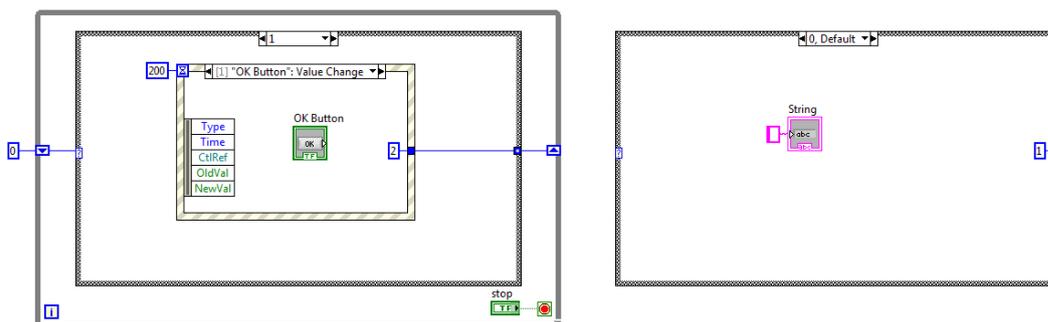


Figura 2.7: Esempio di macchina a stati

2.2 Multisim Connectivity Toolkit

National Instruments prosegue con l'integrazione delle piattaforme NI Multisim e LabVIEW. NI Multisim consente di realizzare lo schema dei circuiti in maniera interattiva, simulandone il comportamento in un ambiente totalmente grafico. Poichè NI Multisim astrae la complessità della simulazione SPICE, non è necessario possedere un'ampia padronanza di SPICE per acquisire, simulare e analizzare rapidamente nuovi progetti. La simulazione integrata riduce la presenza di errori e consente di allestire rapidamente un prototipo con meno iterazioni. Per combinare le funzionalità di simulazione di NI Multisim con la potenza della strumentazione virtuale di LabVIEW e l'acquisizione dati, si deve disporre di uno strumento per condividere i dati tra i due programmi. Per rendere tutto ciò possibile è stato inserito all'interno di LabVIEW un set completo di funzioni che consentono di importare facilmente i dati di simulazione da NI Multisim all'interno di una specifica applicazione LabVIEW e viceversa. Di seguito verranno esposti i più importanti VI di NI Multisim Connectivity Toolkit e di alcuni verranno realizzati esempi o schemi logici.

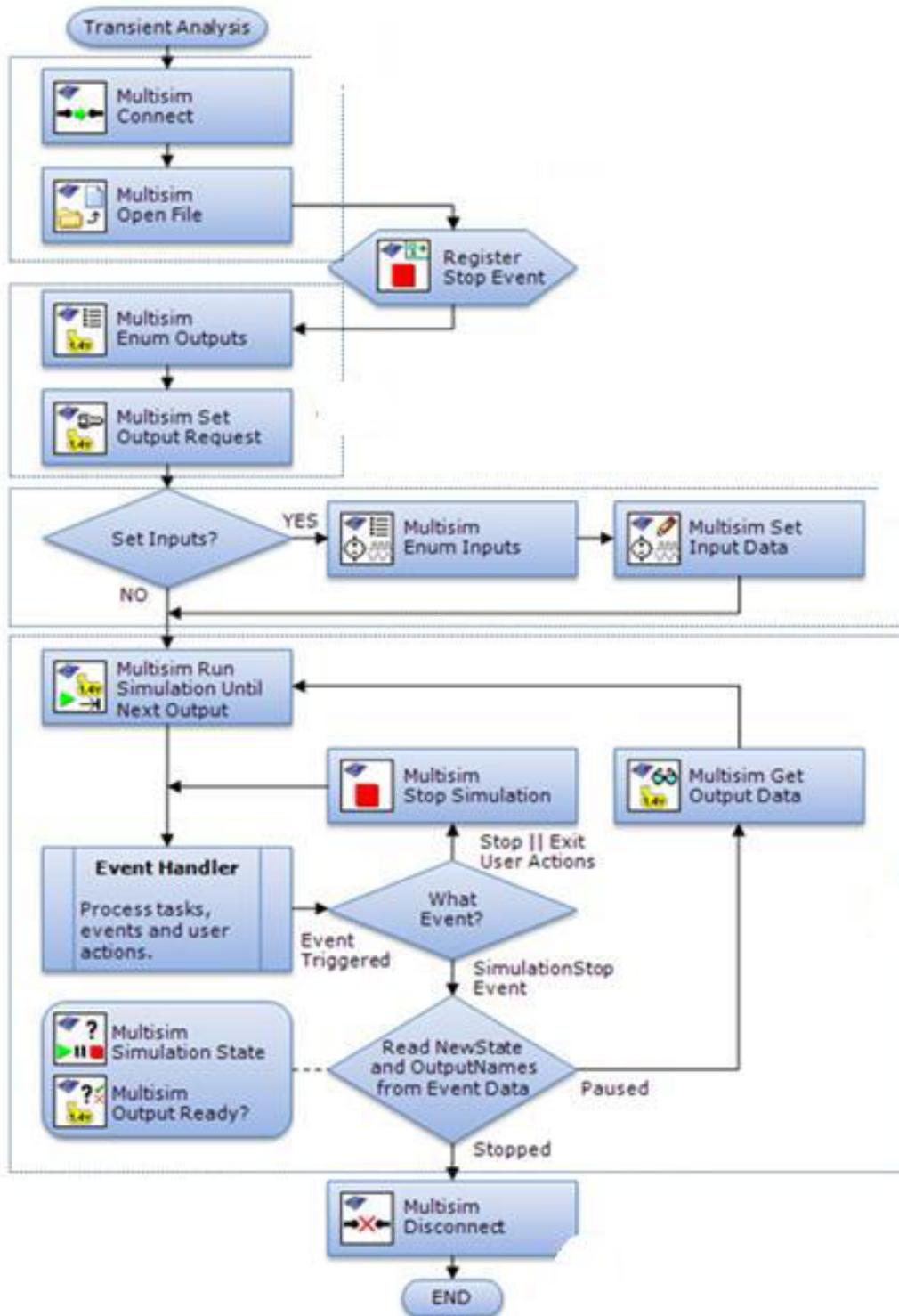


Figura 2.8: Schema logico di procedure ordinate degli strumenti appartenenti a NI Multisim Connectivity Toolkit

2.2.1 I VI della toolkit

Di seguito verranno esposti i più importanti VI di Multisim Connectivity Toolkit e di alcuni verranno realizzati esempi o schemi logici divisi in 5 sezioni².

MULTISIM CONNECTION E MANAGEMENT

Il primo VI è *Multisim Connect.vi*. Questo VI crea un'istanza dell'applicazione NI Multisim. Non fa altro che lanciare NI Multisim come se si facesse un doppio click sull'icona. Quando viene creata quest'istanza non si visualizza nulla nella barra di applicazioni di Windows, solamente perché viene nascosta, anzi, in alcuni calcolatori poco performanti si può vedere l'icona apparire e scomparire quando viene avviato questo VI. Anche nel Task Manager di Windows è possibile visualizzare che NI Multisim è effettivamente in esecuzione. NI Multisim Connect.vi restituisce MultisimApp RefNum cioè il riferimento all'istanza dell'applicazione generata. Questo VI Impiega da 10 a 50 secondi per creare l'istanza di NI Multisim, pertanto, una volta avviato il software, l'utente dovrà stare in attesa.

Multisim Disconnect.vi chiude l'istanza precedentemente aperta, come se si cliccasse sulla x di NI Multisim. Richiede MultisimApp RefNum per essere usato e poter chiudere la relativa istanza. *Multisim Open File.vi* è un VI che permette di aprire un file NI Multisim esistente. Ha bisogno di NI MultisimApp RefNum e del nome path del file da aprire. Equivale in tutto a FILE→open di NI Multisim. Questo VI restituisce l'applicazione che ha utilizzato per essere aperto (MultisimApp RefNum) e MultisimCircuit Refnum cioè il circuito NI Multisim a cui si fa riferimento. Questo è dovuto al fatto che (come succede in NI Multisim) è possibile caricare più circuiti sulla stessa applicazione. *Multisim Save.vi* richiede MultisimCircuit Refnum Out come parametro da passare e sovrascrive il file esistente se impostato su Save, oppure salva su un nuovo file.ms11 se impostato su Save as. In tal caso bisogna, ovviamente, specificare la path di destinazione. *Multisim New File.vi* richiede MultisimApp RefNum per poter creare un nuovo circuito vuoto e restituisce MultisimCircuit Refnum come riferimento a tale circuito.

²La lista dei VI che viene esposta non è completa ma contiene solo i principali VI e quelli utilizzati per TASC e per gli esempi del capitolo 5

MULTISIM CONFIGURATION E CONTROL

I VI che compongono questa sezione di Multisim Toolkit sono sotto-elencati e hanno lo scopo di scegliere, configurare, ripristinare quali saranno gli ingressi e le uscite utilizzate in Labview. *Multisim Enum Input.VI* è costruito per rendere visibile quali sono gli ingressi del circuito NI Multisim utilizzabili da Labview. Questa toolkit permette di utilizzare come ingressi tutti gli AC power e i DC power. Se il circuito NI Multisim non contiene generatori DC o AC power non è possibile gestire gli ingressi. Una volta vista la lista degli ingressi disponibili da Labview si può scegliere che uno, più o tutti gli ingressi sopra elencati siano gestiti da Labview e non più da NI Multisim. Ciò ci permette di riservare dei generatori di NI Multisim al programma Labview consentendo di inserire sul circuito simulato segnali gestiti da Labview. Per fare questo ci si serve del secondo VI di I/O Configuration e Control chiamato *Multisim Reserve Input.VI* e utilizzarlo nella sequenza logica. Dopo aver riservato il generatore per essere utilizzato come ingresso da Labview sarà necessario indicare che segnali inserire sul generatore, altrimenti verrà considerato come generatore nullo.

Multisim Set Input Data.VI è un VI della Toolkit che permette di inserire sul circuito NI Multisim un'array di dati disponibili in Labview. Per poter utilizzare questo oggetto bisogna indicare, mediante una stringa, l'ingresso che è stato riservato, MultisimCircuit Refnum come riferimento al circuito NI Multisim aperto e ovviamente l'array di dati da inserire sul circuito NI Multisim. Sono disponibili 2 modi per inserire il segnale: Sampled e Raw. Con il metodo Sampled bisogna indicare l'array di dati e anche il Sample Rate che indica il numero di campioni al secondo che vengono inseriti nel circuito. Nell'esempio in figura 2.9 viene inserita nel generatore di tensione una rampa che parte da 1 e finisce col valore 4 inserendo 100 valori al secondo. Dato che nell'esempio sopra esposto vengono inseriti 100 campioni al secondo e i valori inseriti sono quattro, allora il generatore produrrà una rampa di tensione che durerà 40 ms. L'inserzione di dati col metodo Raw è del tutto analogo al precedente, ma richiede di avere un'array bidimensionale, dove la prima riga è il riferimento temporale e la seconda il dato numerico. Ciò vuol dire che all'istante i -esimo indicato nella prima riga dell'array bidimensionale il valore istantaneo del generatore sarà quello fornitogli nell'indice i -esimo della seconda riga dell'array. Qualsiasi scelta sia stata fatta sul metodo da utilizzare nell'inserzione di dati questo VI offre la possibilità attraverso il booleano *Repeat*

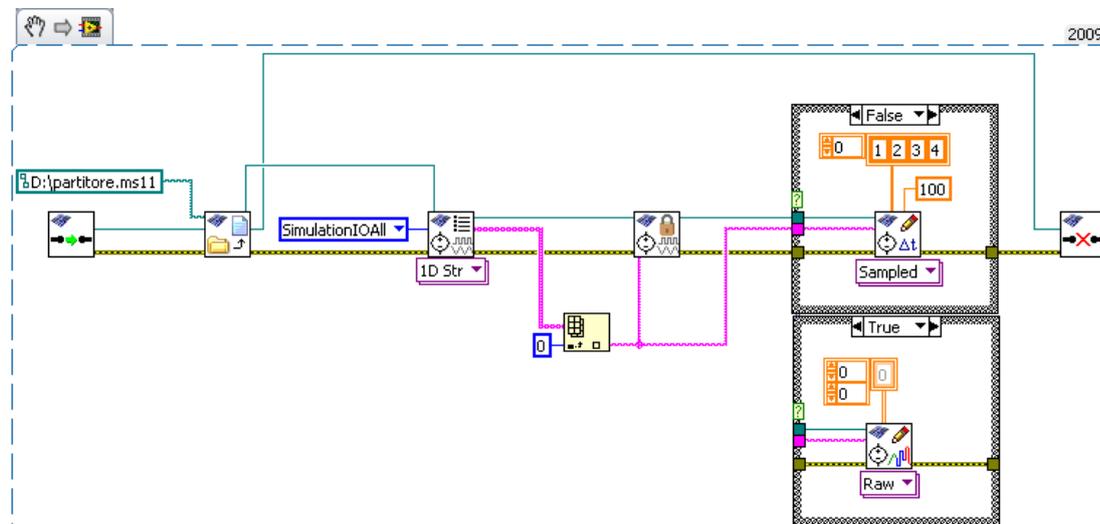


Figura 2.9: Esempio di schema di utilizzo di Multisim Set Input Data, Multisim Connect, Multisim Openfile, Multisim Enum Input, Multisim Disconnect

Data? di poter decidere se ripetere i dati forniti fino alla fine della simulazione. se si utilizza *Multisim Clear Input Data.vi* l'ingrasso V1 torna ad essere gestito da NI Multisim e non più da Labview prolungando il suo effetto per tutta la durata della simulazione. Come logica propone, questo VI deve essere inserito dopo aver riservato un ingresso, altrimenti lo schema di figura 2.9 genererebbe un errore. È anche possibile, cliccando su clear, scegliere di spuntare clear all inputs annullando la gestione di tutti gli ingressi che erano stati riservati e non in singolo ingresso indicato nella stringa. In entrambi i casi il ripristino della gestione degli ingressi non può avvenire durante la simulazione, neppure se essa è un pausa. Tale modifica viene effettivamente applicata quando viene arrestata e poi riavviata la simulazione.

Multisim Enum Output.vi è il VI duale di Multisim Enum Input. Esso fornisce un'array di stringhe. che indicano tensioni e correnti del circuito simulato che è possibile misurare da NI Labview. In particolare si possono osservare tensione e corrente di ogni probe (sonda) presente sul circuito NI Multisim.

Metodi di interpolazione: *SimulationInterpolationRaw* è un'interpolazione arbitraria. Essa garantisce che la durata totale del tempo sarà Number of Sample/Sample, ma la frequenza di campionamento è arbitraria decisa da NI Multisim, il che comporta che il numero di campioni restituito sia molto diverso da quello richiesto si Number of Sample *SimulationInterpolationForceStep* è

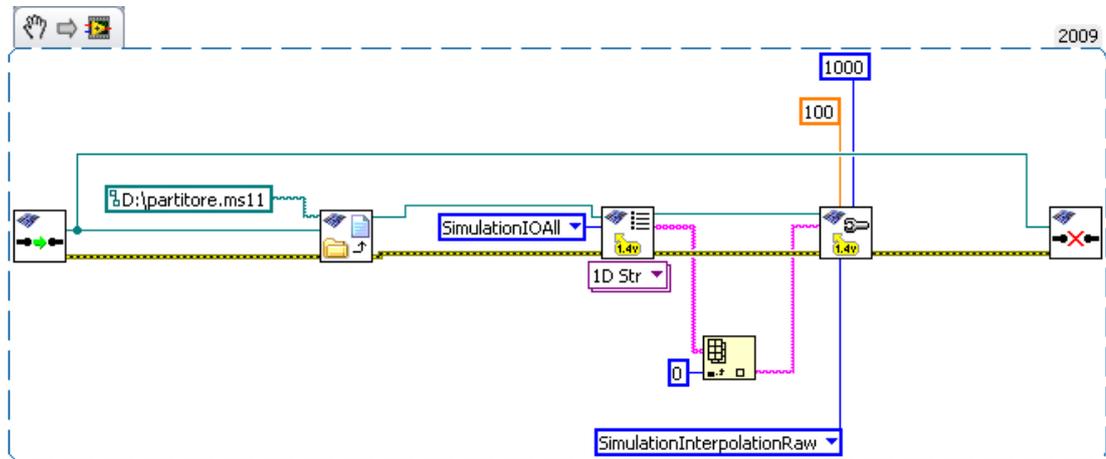


Figura 2.10: Esempio di schema di utilizzo di Multisim Set Output Request

un'interpolazione che forza il simulatore a restituire dati a intervalli regolare pari a $1/\text{Sample Rate}$ per un numero di campioni pari a Number of Sample . Questa interpolazione rispetta perfettamente le richieste di Multisim Set Output Request rispettando esattamente Sample Rate e number of Sample . Se il passo di campionamento naturale di NI Multisim è maggiore di quello che si sceglie su Simple Rate quest'interpolazione rallenta la simulazione, ma se i 2 valori si equivalgono è l'interpolazione che garantisce sincronismo, durata e velocità di esecuzione. $\text{SimulationInterpolationLinear}$ restituisce una semplice interpolazione lineare alla frequenza richiesta e di durata fissata da $\text{Number of Sample}/\text{Sample}$. $\text{SimulationInterpolationSpline}$ a differenza dell'interpolazione polinomiale, che utilizza per approssimare la funzione su tutto l'intervallo di definizione, l'interpolazione spline è ottenuta suddividendo l'intervallo in più intervalli e scegliendo per ciascuno di essi un polinomio di grado più piccolo. Verrà poi impostato che 2 polinomi successivi si saldino in modo liscio garantendo la monotonia. Anche questa interpolazione non fa variare Sample Rate e Number of Sample . $\text{SimulationInterpolationCoerce}$ restituisce un valore di campionamento di uscita pari al valore del campionamento di ingresso più vicino a quello strutturale di NI Multisim. Questo metodo è molto veloce dato che richiede nessun calcolo di interpolazione.

Multisim Output Ready?.vi è un VI che restituisce un valore booleano che indica se i dati per la probe specifica sono pronti. Quando si vogliono graficare n dati di uscita di un circuito bisogna prima avviare la simulazione, attendere la fine della stessa, scegliere l'interpolazione successivamente leggere la risposta del circuito in determinati istanti di campionamento prestabiliti degli n campioni desiderati.

Quindi gli n valori richiesti sono pronti dopo che il simulatore ha eseguito tutti i calcoli necessari. Questo VI restituisce il valore true quando il tutto è stato fatto ed è finalmente possibile graficare o memorizzare gli n dati richiesti. *Multisim Get Output Data.vi* serve per poter finalmente prelevare i dati richiesti quando Multisim Output Ready restituisce il valore true. Ha bisogno di MultisimCircuit Refnum e del nome dell'uscita desiderata. L'istanza creata da Multisim Get Output Data può avere diverse configurazioni: Multisim Get Output Data(Raw 2D DBL) restituisce in uscita un'array bidimensionale $3*n$ dove n indica il numero di campioni. La prima riga è un'array di tempi o frequenze relative all' i -esimo campione. La seconda riga indica la parte reale dell' i -esimo campione e la terza la parte immaginaria dello stesso campione. Se viene fatta un'analisi nel dominio del tempo la terza riga sarà necessariamente nulla. Multisim Get Output Data(Transient 2D DBL) fornisce un'array bidimensionale dove la prima riga è composta dai valori temporali e la seconda dalle ampiezza indicata dalla probe, sia tensioni che correnti. Gli indici delle 2 righe dell'array sono in corrispondenza bi-univoca, il che vuol dire che l' i -esimo componente della prima riga indica l'istante in cui l'ampiezza in considerazione vale quanto indicato nell' i -esimo componente della seconda riga. Multisim Get Output Data(Transient XY) si comporta esattamente come il *Transient 2D DBL*, ma non fornisce un'array bidimensionale, bensì un cluster contenente 2 array, la prima contiene i valori temporali, la seconda le rispettive ampiezze. Multisim Get Output Data(Transient Wfm) restituisce i dati della simulazione come un oggetto Waveform che contiene l'array di lavoro, il valore dell'istante iniziale e il dt che corrisponde al tempo tra un valore e il successivo. Multisim Get Output Data(AC Sweep 2D DBL) è la configurazione che fornisce la risposta del circuito sottoposto ad uno sweep in frequenza. Questo tipo di configurazione verrà trattato più approfonditamente in seguito, quando verrà esposto le funzionalità un altro VI della toolkit che permette lo sweep in frequenza. Per ora osserva solo che la configurazione AC Sweep 2D DBL fornisce un'array $3*n$, dove n indica il numero di dati e le tre righe sono rispettivamente le frequenze, le parti reali e infine, le parti immaginarie delle rispettive frequenze. Multisim Get Output Data(AC Sweep Freq Re Img) restituisce tre singoli array contenenti i valori delle frequenze, delle parti reali e delle parti immaginarie. Multisim Get Output Data(AC Sweep Freq Db Deg) restituisce tre array, un'array unidimensionale i cui valori sono le frequenze in cui è stato fatto lo sweep, un'ar-

ray del modulo in Bd e un'altra array della fase della risposta in gradi(degrees). Nella scelta della configurazione di Multisim Get Output Data, sarà fondamentale considerare se si lavora nel dominio del tempo o delle frequenze. Una volta scelto il dominio utilizzare una configurazione o l'altra cambia solo il modo in cui ci vengono forniti i dati richiesti. Quindi la scelta giusta da fare dipenda da cosa si deve fare con i dati finalmente ricavati. L'ultimo VI di I/O Configuration

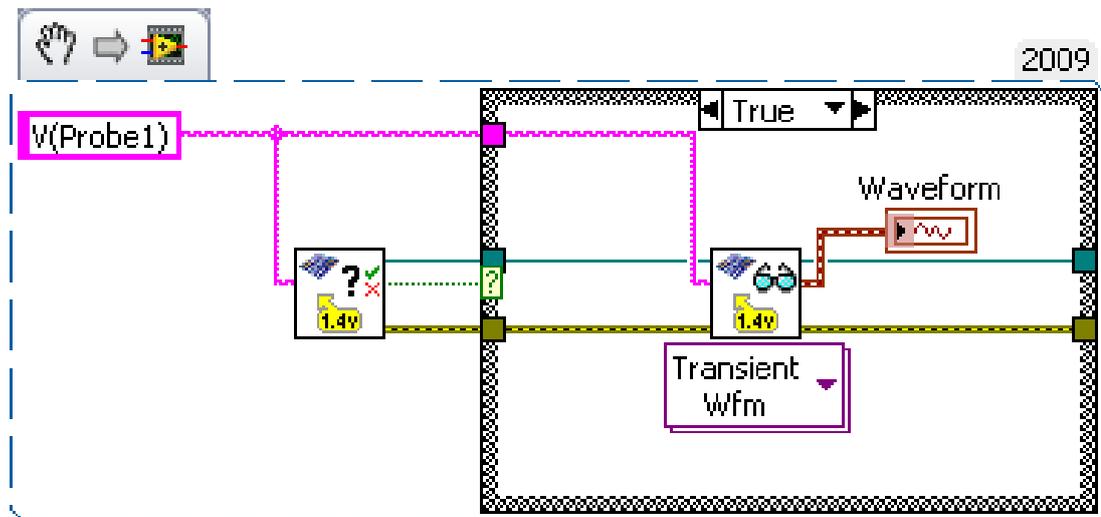


Figura 2.11: Esempio di schema di utilizzo di Multisim Output Ready

e Control si chiama *Multisim Clear Output Request.vi* che si comporta analogamente a Multisim Clear Input Data. In questo caso il VI cancella, o annulla, una precedente richiesta di uscita per una sonda specifica, oppure cancella tutte le richieste di uscita fatti sino a qual momento.

SIMULATION CONTROL

Questo gruppo di VI permette di gestire la simulazione spice di NI Multisim attraverso Labview. Sono dei VI che permettono di avviare, arrestare, mettere in pausa la simulazione di NI Multisim. Il primo preso in considerazione è *Multisim Run Simulation.vi* che permette di far eseguire la simulazione. Non è possibile fare nessuna analisi se non è stata eseguita la simulazione, non è possibile recuperare dati dal circuito NI Multisim se non è stato eseguito il circuito stesso. Multisim Run Simulation richiede MultisimCircuit Refnum come riferimento al circuito da simulare e accetta in ingresso anche un valore numerico (che di default è 1) con cui è possibile decidere per quanti secondi far eseguire la simulazione. In seguito si vedrà, grazie ad alcuni esempi, che la durata della simulazione è un

parametro fondamentale per il corretto funzionamento. Multisim Run Simulation offre anche la possibilità di scegliere un valore booleano che indica se alla fine della simulazione deve verificarsi un evento pausa o stop. Anche questa scelta deve essere ben ponderata considerando che in pausa non è possibile né salvare il circuito NI Multisim, né cambiare parametri dei componenti. D'altra parte mettere in pausa la simulazione e poi riprenderla vuol dire continuare la simulazioni con condizioni iniziali non necessariamente nulla. Se la simulazione viene stoppata e poi avviata di nuovo le condizioni iniziali di componenti come condensatori e induttori saranno nulle. Ciò non avviene se viene messa in pausa e poi ripresa la simulazione. *Multisim Run Simulation Until Next Output.vi* risolve molti dei problemi della gestione dello stato della simulazione. Avvia la simulazione fino a che, per ogni istante multiplo di dt esistono uscite valide e poi mette la simulazione in pausa. Questo VI è estremamente comodo e si deve usare in sostituzione di Multisim Run Simulation. In questo caso non serve indicare quanto tempo deve durare la simulazione. Questo VI eredita da Multisim Set Output Request la frequenza di campionamento e sa che deve mettere in pausa la simulazione quando ad un istante multiplo di dt acquisisce un'uscita non valida. In tal caso mette in pausa la simulazione. Questo modo di procedere all'acquisizione dati ha però un problema: Quando le risorse della macchina su cui lavorano Labview e NI Multisim sono poche, oppure molto impegnate in altri fronti, oppure ancora i circuiti Multisim in esame sono molto sofisticati, si rischia che il sistema operativo perda in qualche modo il sincronismo temporale tra Labview e Multisim dando, all'istante multiplo di dt , un valore non valido in uscita. Multisim Run Simulation Until Next Output valuta tale campione come primo da non acquisire e termina la simulazione mettendola in pausa anche se non è effettivamente l'ultimo. L'utilizzo di Multisim Run Simulation richiede un po di attenzione sul booleano *paused when finished?* e sul numero *Seconds*, ma garantisce sempre la corretta durata della simulazione. Un altro strumento importante è *Multisim Wait For Next Output.vi* che è di fondamentale importanza in molti casi perché attende fino a che l'uscita è disponibile oppure è terminata la simulazione. La sequenza logica da seguire prima di avviare la simulazione è di riservare prima tutti gli ingressi e le uscite, avviare la simulazione e successivamente attendere fino a quando sono disponibili dati della simulazione. NI Multisim richiede una certa quantità di tempo per eseguire i calcoli e poter fornire i dati richiesti. Questa è

la funzionalità di Multisim Wait For Next Output . Senza questo polimorfismo verrebbero richiesti i dato in uscita prima che essi siano disponibili generando un errore.

CIRCUIT CONFIGURATION

Il primo VI che verrà preso in considerazione all'interno di Circuit Configuration prende il nome di *Multisim Enum Components.vi* e svolge il semplice, ma utile ruolo di elencare tutti i componenti presenti nel circuito, il VI restituisce i componenti in forma di stringa, oppure in forma di variant. È possibile decidere di utilizzare un filtro per elencare solo determinate categorie di componenti. Component All: vengono elencati tutti i componenti; Component AllRLC: Vengono elencati solo resistenze, condensatori e induttanze; Component Resistor: L'array contiene tutte e sole le resistenze del circuito; Component Capacitor: L'array contiene tutti e soli i condensatori del circuito; Component Inductor: L'array contiene tutte e sole le induttanze del circuito.; Un altro utile VI presente nella toolkit è *Multisim Replace Component.vi* che serve per sostituire un componente presente nel circuito NI Multisim con un altro presente nelle librerie. Ovviamente richiede MultisimCircuit Refnum come riferimento al circuito, ma richiede anche una serie di parametri per la sostituzione. Si supponga di voler sostituire, nello schema in figura 2.12, l'amplificatore operazionale con un comparatore. Lo schema Labview

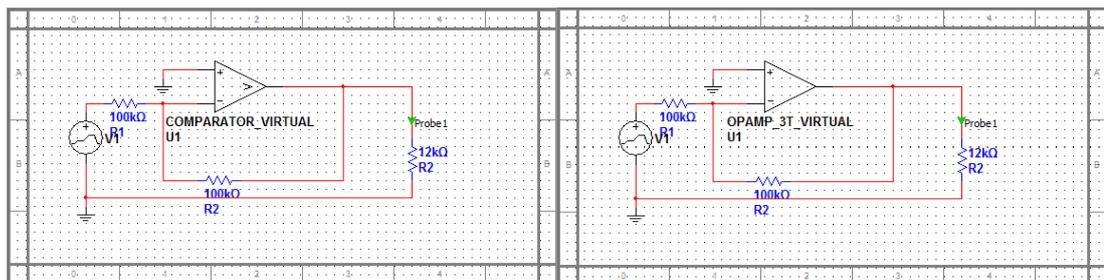


Figura 2.12: Circuito Multisim con Amplificatore operazionale che verrà sostituito da un comparatore

di figura 2.13 utilizza Multisim Replace Component si richiedono 7 parametri, 5 dei quali sono quelli cerchiati in figura 2.12. Component RefDes: è la refdes, ovvero il nome, del componente che si vuole rimpiazzare (nell'esempio sarà sostituito U1); Section: è il nome della sezione se si vuole sostituire un componente all'interno di una multisessione, come visto per Multisim Enum Section. Se il componente non appartiene a nessuna sezione questa stringa rimarrà vuota(come

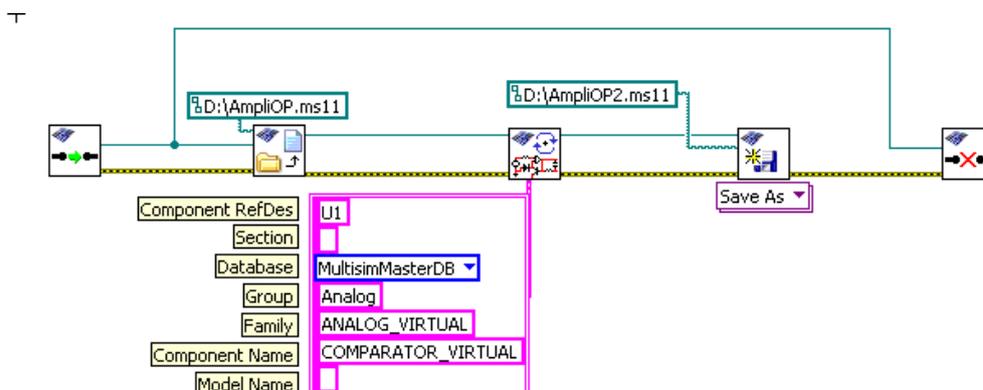


Figura 2.13: Circuito Multisim con amplificatore operazionale sostituito al comparatore

nel esempio preso in considerazione); Database: Indica su che Database è presente il nuovo componente (MultisimMasterDB); Group: è il gruppo di database dove è presente il comparatore (analog); Family: è la famiglia del database (ANALOG_VIRTUAL); Component Name: è il nome del componente (COMPARATOR_VIRTUAL); Model Name: è il nome del modello del componente. Se il nuovo componente ha più modelli, questa stringa specifica quale utilizzare. Se non viene specificato nulla (come nel esempio preso in considerazione) sarà preso il primo modello di default. Questo VI lancia un errore sia se Component RefDes e Section del componente da sostituire non esistono, sia se non esistono i 5 parametri del nuovo componente, sia nel caso si tenti di sostituire un componente mentre la simulazione è in corso. Se non si verifica nessuno di questi errori viene restituita una stringa che indica la RefDes del nuovo componente. Multisim Replace Component non permette di rimpiazzare componenti RLC, anche se come si vedrà in seguito sarà possibile variarne il valore. Ciò vuol dire che una resistenza rimarrà sempre una resistenza, un condensatore rimarrà condensatore e un'induttanza resterà un'induttanza. Come si può procedere se si vogliono cambiare i valori dei componenti RLC? Il seguente schema, che utilizza *Multisim RLC Value.vi*, permette questa funzione estremamente utile. Multisim RLC Value può essere impostato su Set oppure si Get. Nella figura 2.15 vengono utilizzate entrambe le configurazioni. Precisamente viene passato come parametro la RefDes del componente in considerazione e la configurazione Get restituisce l'attuale valore. Successivamente la configurazione Set, che riceve essa la stessa Refdes, e imposta

il nuovo valore del componente.



Figura 2.14: Nuove impostazioni dei valori resistivi

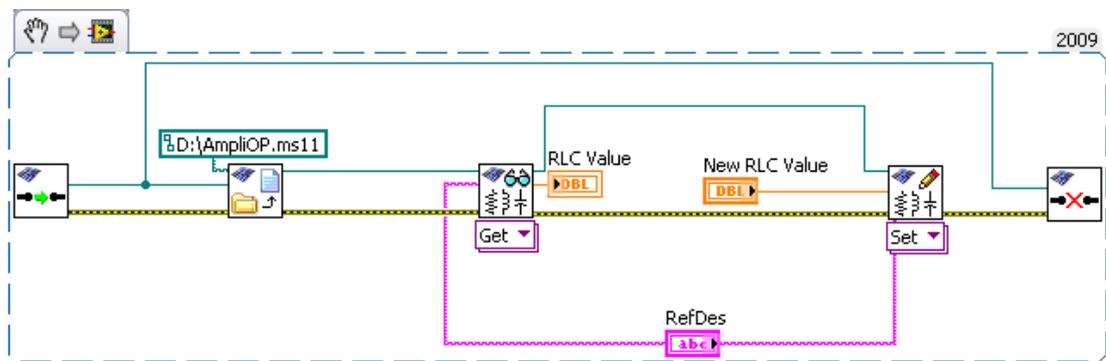


Figura 2.15: Schema multisim che utilizza Multisim Set RLC Value

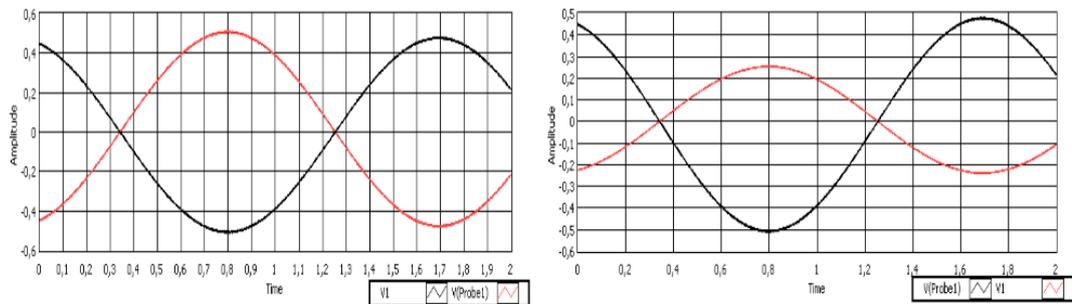


Figura 2.16: Uscita del partitore con i nuovi valori RLV

ERROR AND UTILITY

Multisim Last Error Message.vi ha la funzionalità di poter visualizzare da Labview gli errori generati da NI Multisim. Può gestire sia gli errori provocati dall'applicazione, sia quelli generati dal singolo circuito. L'analisi degli errori può essere fondamentale per poter capire l'origine degli stessi e l'eventuale correzione. In oltre è di fondamentale importanza separare gli errori dell'applicazione con gli errori del singolo circuito. Questo VI restituisce sia il testo generato dall'errore sia un booleano che identifica la presenza dell'errore facilitandone l'eventuale

gestione in Labview. Come visto in precedenza esistono dei VI che compongono la toolkit che, se non correttamente configurati, possono generare altri errori. Sapere l'origine degli errori, sapere che provengono da una configurazione non buona dei VI, oppure da una pessima gestione del circuito NI Multisim oppure ancora dall'applicazione stessa NI Multisim è di fondamentale importanza.

2.2.2 Schema a blocchi minino

L'utilizzo dei VI della toolkit permette di migliorare il processo di progettazione di un circuito elettronico. La capacità di eseguire simulazione, gestire dati numerici, elaborare e visualizzare i risultati, modificare parametri della simulazione, avviare la simulazione con nuovi parametri dei componenti del circuito simulato consente una progettazione ad hoc del circuito. Nel prossimo capitolo verranno studiate la potenzialità di LabView nell'acquisizione di dati reali trasformandoli in valori numerici che possono essere utilizzati come segnale reale da utilizzare nella simulazione. L'idea è quella di poter collaudare un circuito virtuale con un segnale uguale a quello a cui sarà realmente utilizzato il circuito simulato fornendo dei risultati reali e non più ideali dato che il segnale d'ingresso è reale. Multisim Connectivity Toolkit sfrutta la simulazione circuitale SPICE per collaudare il circuito in esame utilizzando le strutture di un linguaggio ad alto livello come i cicli e le strutture decisionali. In figura 2.18 è mostrato lo schema a blocchi di labview che permette di collaudare il circuito partitore2.ms11 mostrato in figura 2.17.

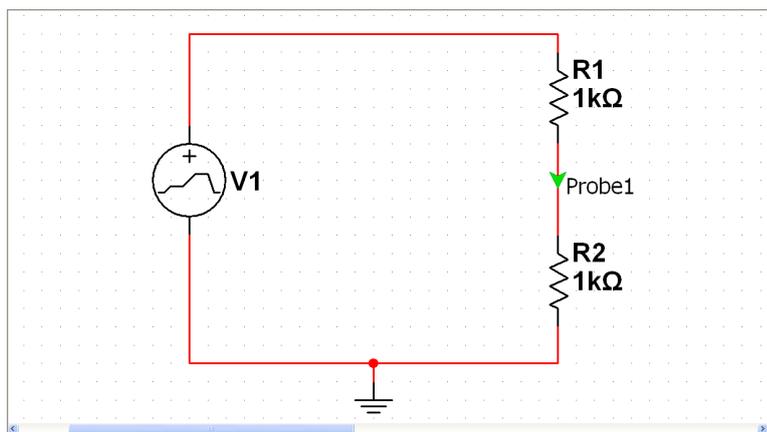


Figura 2.17: Semplice esempio di circuito multisim con partitore

Il semplice schema partitore è stato scelto solo a titolo di esempio per mostrare in figura 2.19 la logica di questa procedura di collaudo dove viene mostrato

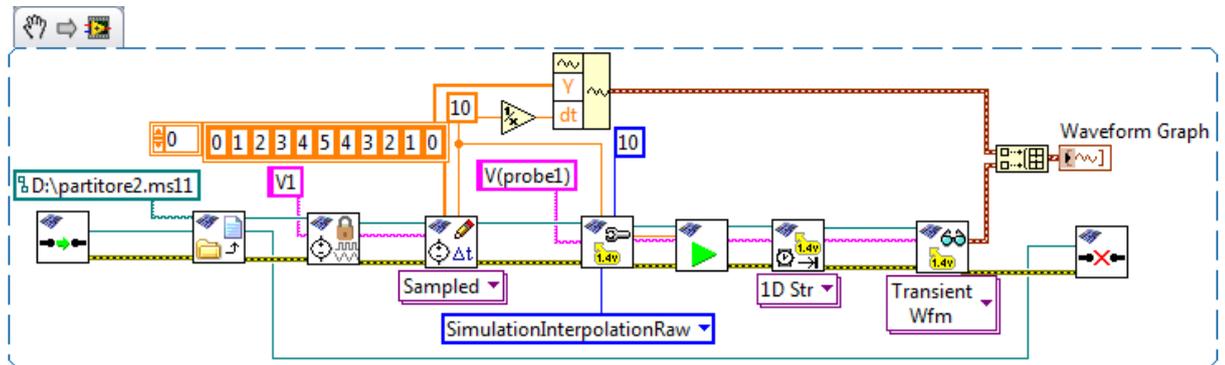


Figura 2.18: Schema minimo per la simulazione circuitale attraverso Labview

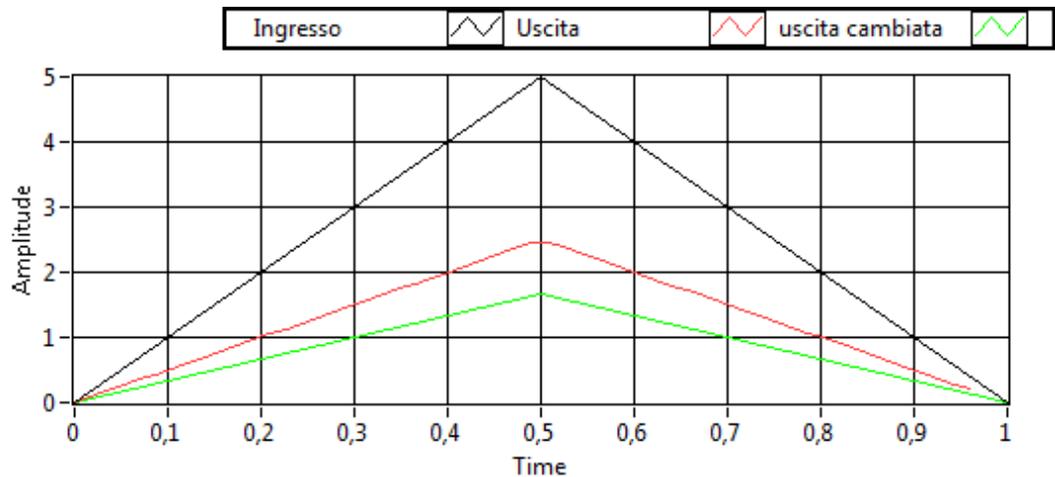


Figura 2.19: Ingresso (nero); uscita per $R_1 = R_2$ (rosso); uscita per $R_1 = 2R_2$ (verde)

sia l'ingresso sia l'uscita. Nello schema a blocchi sono stati inseriti 11 valori (0,1,2,3,4,5,4,3,2,1,0) intervallati tra loro da 0,1 secondi. Questo rappresenta il segnale d'ingresso che definisce l'evoluzione temporale e delle ampiezze del generatore V1. Il grafico in rosso mostra il valore della tensione misurata dalla probe1 utilizzando come ingresso il generatore di tensione. Lo schema a blocchi di figura 2.18 mostra la logica sequenziale delle istruzioni seguite da labview. In questo modo viene collaudato il circuito partitore2.ms11 con dati inseriti da labview e non più con segnali ideali di NI Multisim. Nel grafico è presente una terza traccia verde che rappresenta l'andamento della tensione di probe1 cambiando il valore della resistenza R1 da $1\text{k}\Omega$ a $2\text{k}\Omega$ utilizzando Multisim Ser RLC Value che non è stato inserito nello schema a blocchi per non appesantire il disegno. Il

2. NI LABVIEW

test del circuito può essere ripetuto iterativamente grazie alle strutture Labview e visualizzare il comportamento del circuito per i diversi valori dei componenti senza aprire NI Multisim, ma solo utilizzando labview e Multisim Connectivity Toolkit.

Capitolo 3

Conversione A→D e schede di acquisizione (ELVIS)

3.1 Conversione analogica-digitale

3.1.1 Il campionamento

Il processo di campionamento è il primo passo per la conversione di un segnale da analogico a digitale. Il semplice concetto che nasconde questo importante procedimento è quello di valutare, a intervalli regolari di tempo, il valore istantaneo del segnale da convertire. Procedendo in questo modo si ottengono tanti valori numerici quanti sono gli istanti valutati nel campionamento. L'insieme di questi valori numerici prende il nome di *sequenza di campioni* e l'intervallo di tempo tra un campione ed il successivo viene chiamato *Tempo di Campionamento* o T_c . Analogamente viene definita la *Frequenza di campionamento* o *Sample Rate* il reciproco del tempo di campionamento

$$f_s = \frac{1}{T_c}$$

e definisce il numero di campioni per unità di tempo. La frequenza di campionamento è uno dei parametri più importanti nella conversione di un segnale analogico. Acquisire ad alta frequenza equivale a prendere i campioni ad istanti di tempo molto vicini tra loro trascurando meno informazioni riguardante il segnale originale. Il circuito che effettua il campionamento può essere pensato come un interruttore elettrico pilotato da un segnale di clock. Di seguito si studierà

3. CONVERSIONE AD E SCHEDE DI ACQUISIZIONE (ELVIS)

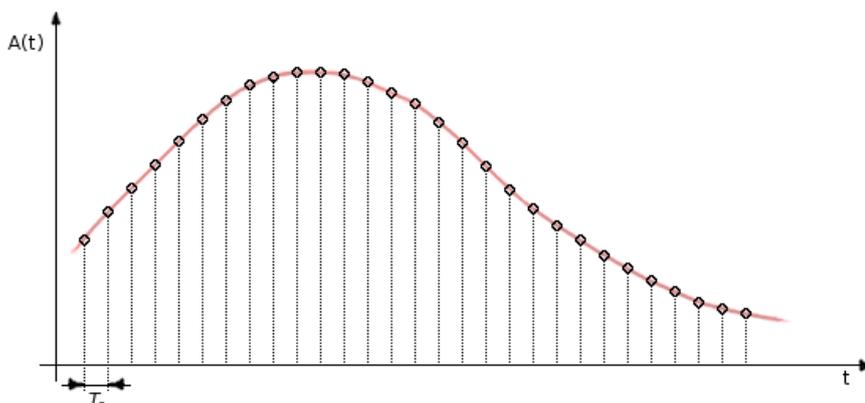


Figura 3.1: Campionamento di un segnale continuo

che la frequenza di campionamento è un parametro che definisce anche il tipo di segnale che può essere acquisito. Il teorema del Nyquist afferma che, sotto le opportune ipotesi, la minima frequenza di campionamento necessaria per evitare ambiguità nella ricostruzione del segnale con larghezza di banda finita e nota è pari al doppio della sua massima frequenza.

Teorema di Nyquist: Un segnale $s(t)$ a banda limitata da f_M può essere univocamente ricostruito dai suoi campioni $s(n\Delta t)$ $n \in \mathbb{Z}$ presi a frequenza $f_s = \frac{1}{\Delta t}$, se $f_s > 2f_M$

Se il teorema del campionamento non viene rispettato il segnale originale non può essere ricostruito a partire dai campioni perché tali campioni non possono ricostruire univocamente un unico segnale

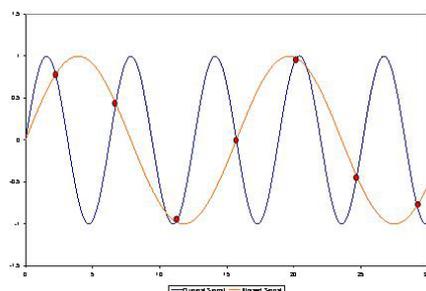


Figura 3.2: Fenomeno di aliasing nel dominio del tempo

Il campionamento di un segnale nel dominio del tempo ha come effetto, nel dominio della frequenza, la ripetizione periodica della sua trasformata di Fourier a

multipli di f_s .

Nelle figure 3.2 e 3.3 vengono mostrati fenomeni di Aliasing, cioè effetto dovuti ad una frequenza di campionamento che non rispetta il teorema di Nyquist. Per proteggere l'acquisizione da questo fenomeno indesiderato si è soliti usare un filtro anti-aliasing che è composto da un filtro passa-basso che elimina le frequenze del segnale d'ingresso che potrebbero generare aliasing.

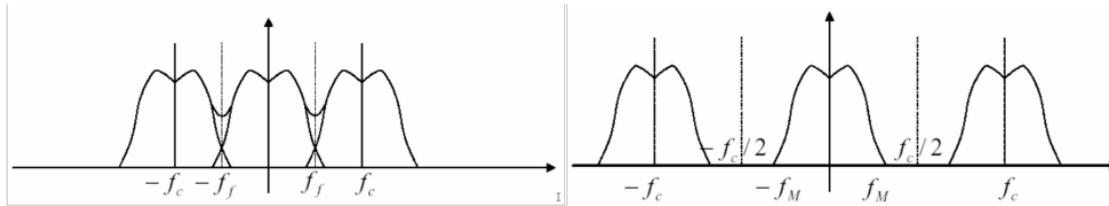


Figura 3.3: Fenomeno di aliasing nel dominio della frequenza

3.1.2 La quantizzazione

La quantizzazione è l'operazione attraverso la quale il segnale campionato presente all'uscita del campionatore è trasformato in una sequenza di livelli di tensione quantizzati. In altre parole il campionatore arrotonda il valore numerico campionato al più vicino livello di quantizzazione. La figura 3.4 mostra un esempio di segnale campionato e quantizzato a livelli multipli di q . Q (oppure q) è chiamato **passo di quantizzazione**

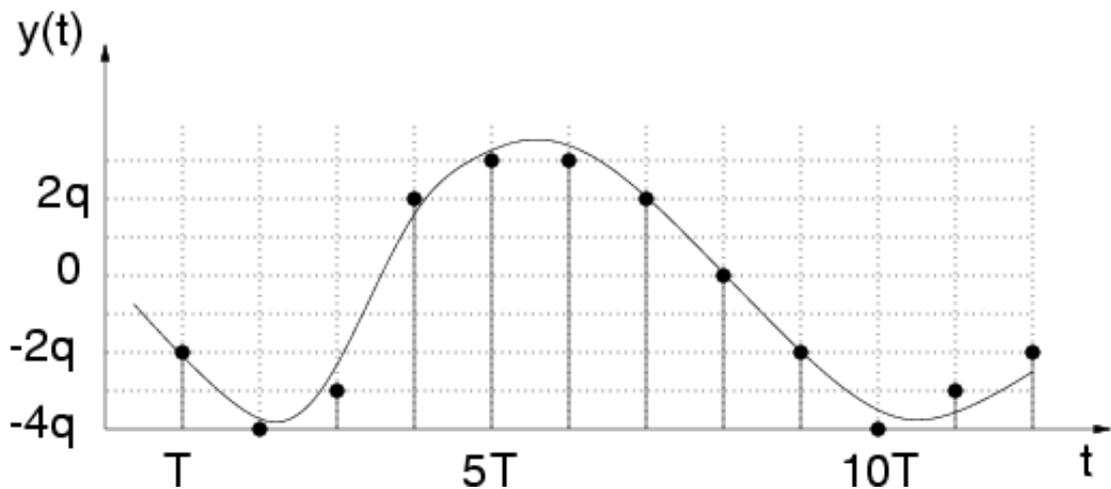


Figura 3.4: Esempio di segnale campionato e quantizzato

3. CONVERSIONE AD E SCHEDE DI ACQUISIZIONE (ELVIS)

Q è definito come

$$q = \frac{R}{N}$$

dove R indica l'intervallo di valori numerici accettabili in ingresso ed N il numeri di passi di quantizzazione che. N, a sua volta, è definito come:

$$N = 2^B$$

dove B indica il numero di bit. Il che comporta che

$$q = \frac{R}{2^B}$$

. Se si prende in considerazione l'esempio di figura 3.5 si può capire che il segnale che deve essere quantizzato deve appartenere ad un intervallo di valori che in convertitore ammette. Nell'esempio di figura l'intervallo è (0; 8). Sempre nella stesse figura vengono mostrati 8 codici che corrispondono a 8 livelli di quantizzazione che si ottengono con 3 bit ($8 = 2^3$). Il quantizzatore caratterizzato dalla transcaratteristica di figura 3.5 fornirà il codice 000 per tutti i campioni appartenenti all'intervallo [0; 1), fornirà il codice 001 per tutti i campioni appartenenti all'intervallo [1; 2), ecc. Il convertitore fornirà 000 anche per tutti i valori dei campioni inferiori a 0 e fornirà 111 per tutti i campioni maggiori di 8.

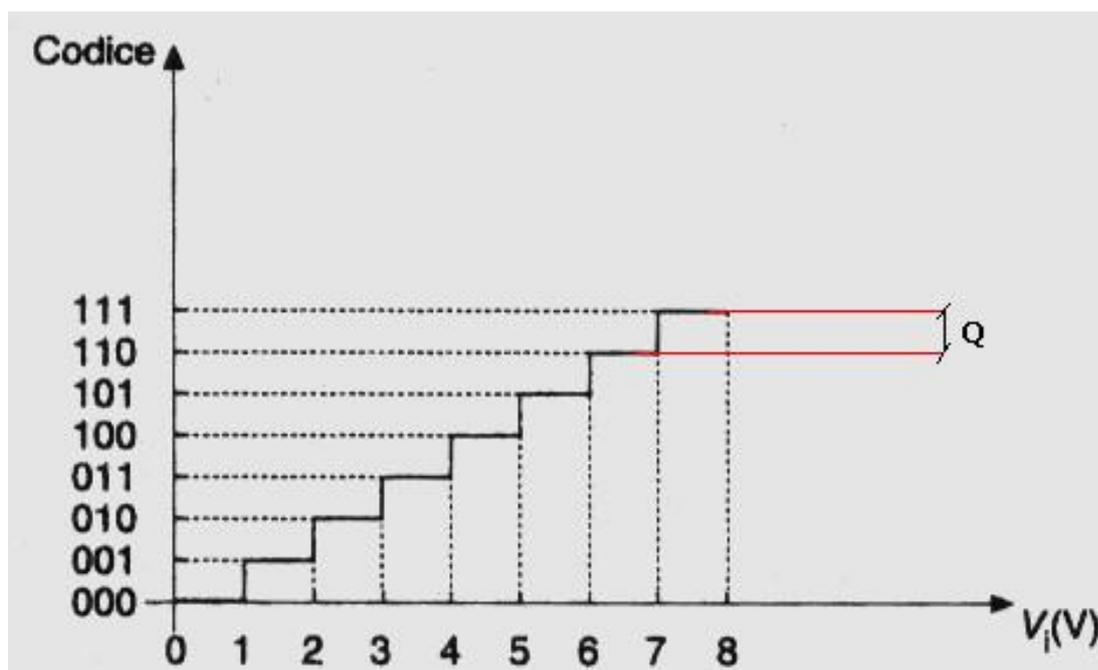


Figura 3.5: transcaratteristica di un quantizzatore

Date queste definizioni e mostrati questi esempi è facile intuire che la quantizzazione è caratterizzata da due elementi che sono: i bit (B) e dall'intervallo dei valori in ingresso (R) che assieme definiscono q . Si fa notare che, in questo caso, q definisce anche l' **errore di quantizzazione** che identifica, ad ogni istante di campionamento, la differenza massima tra il valore quantizzato ed il valore del campione.

Tutte queste informazioni servono a capire che la conversione di un segnale deve essere eseguita facendo attenzione alla frequenza di campionamento, ai bit di quantizzazione e al range di valori in ingresso. Questi tre parametri definiscono il grado di approssimazione del segnale originale con quello campionato e quantizzato chiamato **risoluzione del convertitore $A \rightarrow D$** .

National Instruments è l'azienda leader al mondo nell'acquisizione dati e nella conversione $A \rightarrow D$ e $D \rightarrow A$ e fornisce una vastissima gamma di schede che eseguono campionamento, quantizzazione, filtraggio ed adattamento in scala per ottimizzare la risoluzione¹. Tutte queste schede sono completamente compatibili con Labview. Il software e gli hardware appena presentati permettono l'analisi dati reali attraverso un completo linguaggio di programmazione grafica e consentono la realizzazione di sistemi di acquisizione ed elaborazione dati molto precisi se correttamente progettati.

3.2 Schede di acquisizione (ELVIS)

Le schede di acquisizione di National Instruments sono moltissime, ma per la progettazione e lo sviluppo di TASC e lo studio di Multisim Connectivity Toolkit è stata utilizzata una scheda di acquisizione particolare che si chiama NI ELVIS.

National Instruments Educational Laboratory Virtual Instrumentation Suite (NI ELVIS) è una suite di 12 strumenti di laboratorio che comprende anche: scheda di acquisizione, generatore di forme d'onda arbitrarie, acquisizione e generazione

¹Le schede di acquisizione da National Instruments sono in grado di utilizzare l'intero range del quantizzatore per ottimizzare la risoluzione e lasciare invariato l'errore. Questo si ottiene amplificando il segnale d'ingresso tale da far sì che occupi l'intero range per poi dividerlo per lo stesso coefficiente all'uscita del quantizzatore

3. CONVERSIONE AD E SCHEDE DI ACQUISIZIONE (ELVIS)

digitale, multimetro, oscilloscopio, analizzatore di Bode, impedenziometro, alimentatore stabilizzato di tensione. ELVIS è composto da una struttura di base che contiene tutti gli strumenti e diverse schede compatibili per i diversi usi di NI ELVIS; quella utilizzata per lo sviluppo di TASC è la scheda base che comprende una breadboard su cui è possibile inserire fisicamente i piedini di componenti elettronici. Tale scheda offre la possibilità di costruire un circuito elettronico che può essere facilmente alimentato e studiato grazie agli strumenti integrati in ELVIS. ELVIS può essere utilizzato anche come semplice scheda di acquisizione e generazione. Di seguito verranno riportate alcune caratteristiche presenti sul data-sheet:

- Number of input channels 8 differential or 16 single ended
- ADC resolution 16 bits
- Sample Rate 1.25 MS/s single channel
- Input Impedance $> 10G\Omega$
- Input range ± 10 V
- Number of output channels..... 2
- Maximum update rate..... 2.8 MS/s
- Output range ± 10 V
- Maximum output current..... 500 mA
- Output impedance..... 1Ω

Questi dati servono a poter caratterizzare i segnali che possono essere acquisiti. Possono essere acquisiti 8 canali differenziali con una frequenza di campionamento di circa 165 Hz per canale, se vengono acquisiti 8 canali, oppure permette un'acquisizione a 1.25 MHz in un singolo canale. Questi dati permettono anche di definire le frequenza massime dei segnali d'ingresso che devono rispettare il teorema di Nyquist. Con ELVIS è quindi possibile acquisire segnali con frequenza massima minore di

$$f_s \leq 625kHz$$

. Allo stesso modo il numero di bit e il range d'ingresso identificano l'errore massimo di quantizzazione pari a

$$q = \frac{R}{2^B} = \frac{20}{2^{16}} \sim 0.0000305V$$

. Questo valore identifica sia l'ampiezza minima che deve avere il segnale per essere rilevato in ingresso, sia l'errore massimo di quantizzazione. Per una corretta conversione il segnale deve appartenere all'intervallo

$$R = [-10; +10]V$$

. Per acquisire un segnale che non rispetti le caratteristiche fisiche ELVIS non può essere utilizzato, quindi è necessario utilizzare schede di acquisizione che abbiano un'architettura diversa. Nello sviluppo del software di collaudo di Multisim Connectivity Toolkit è stato utilizzato ELVIS per acquisire i segnali, per cui tutte le misure e le osservazioni fatte nel capitolo 5 sono riferite a tale strumento e alla sua struttura hardware. ELVIS è stato utilizzato anche come scheda di generazione di segnali analogici a partire da variabili Labview, quindi anche i segnali generati hanno ovviamente delle restrizioni riguardanti le frequenze, le ampiezze massime e l'approssimazione del segnale rispetto al corrispondente analogico.

Capitolo 4

TASC

TASC (Test Analysis Simulated Circuits) è un software Labview che utilizza buona parte degli strumenti di Multisim Connectivity Toolkit per collaudare le funzionalità, i pregi e i difetti dell'integrazione tra la simulazione virtuale e gli strumenti di analisi Labview. TASC offre all'utente una comoda interfaccia grafica che permette lo studio dei circuiti simulati con dati reali catturati grazie all'acquisizione delle schede prodotte da National Instruments. Il programma è stato progettato per poter gestire un solo ingresso e ricavati i dati da una sola uscita del circuito NI Multisim. TASC è costituito da due parti: le tre grandi tab (Multisim Circuit, Data acquisition e Data View) e l'indicatore a destra dove è possibile vedere lo stato attuale del software, riavviarlo e arrestarlo. TASC è stato creato utilizzando circa 700 VI strutturati in una macchina a stati composta da: stati iniziali sequenziali, nei quali viene configurato il circuito NI Multisim; stati centrali, nei quali viene eseguita l'acquisizione e la simulazione; stati finali di analisi eseguibili con la pressione di alcuni pulsanti presenti sul pannello frontale.

4.1 Circuito Multisim

Appena avviato il programma viene lanciato Multisim Connect.vi che apre un'istanza di NI Multisim. Questa procedura richiede alcuni secondi in base alla potenza del calcolatore su cui gira. Durante quest'operazione il cursore del mouse viene disabilitato e sostituito dalla classica clessidra di Windows. Il passo successivo sarà decidere quale circuito NI Multisim analizzare e collaudare con TASC. Fino a che non è stato selezionato un file.ms** la path dei file, chiamata MS file,

L'output l'utente dovrà selezionare il punto del circuito nel quale inserire i valori numerici che caratterizzano il segnale reale acquisito. Nel circuito in esame è presente uno solo tra i generatori alternati o continui di tensione o corrente, quindi l'utente potrà selezionare l'unica opzione esistente tra quelle proposte nel menù a tendina. Il software è stato costruito per rendere l'utilizzo facile all'utente, per questo i passi da seguire devono essere cronologicamente ordinati e indicati dal lampeggio del comando, che deve essere utilizzato in ogni istante, bloccando ogni altro controllo. Così facendo sarà impossibile per l'utente dare istruzioni non previste al software e precedere con l'analisi in maniera semplice e guidata. L'indicatore a destra mostra passo passo il punto in cui è fermo l'operatore, i passi successivi e i passi precedenti svolti correttamente. In caso di un errore da parte dell'utente è possibile, in ogni istante, ripartire dal punto iniziale con la scelta del circuito NI Multisim e scegliere nuovamente l'ingresso e l'uscita. La procedura di reset consente di ripartire dopo aver creato l'istanza di NI Multisim facendo risparmiare all'utente il tempo di connessione. Se invece viene premuto il pulsante di STOP il software elimina l'istanza di NI Multisim precedentemente creata ed chiude TASC. Avviare nuovamente il software richiede ancora l'attesa del tempo necessario alla creazione della nuova istanza NI Multisim.

4.2 Acquisizione e campionamento

Dopo aver configurato correttamente la sezione Multisim Circuit, TASC passa a visualizzare la sezione Acquiring mostrata in figura 4.2.

Questa sezione serve ad acquisire tramite scheda di acquisizione il segnale reale. Il comando in alto a sinistra *Simulated o Real* permette all'utente di scegliere se far provenire il canale da un device di acquisizione reale oppure da uno simulato. Tale opzione è stata inserita per poter far collaudi veloci e analisi anche in assenza di un hardware esterno. Dopo aver deciso il device, TASC carica sul menù a tendina tutti i canali analogici disponibili per l'acquisizione. A questo punto l'utente deve selezionare il canale desiderato, TASC mostra nel display un grafico che identifica il segnale presente nel canale selezionato. Mentre il segnale viene visualizzato l'utente può decidere di cambiare *device* o *canale* o *frequenza di campionamento* o *Number of Samples*. In questo caso il segnale visualizzato cambierà in base alle nuove istruzioni fornite. Quando la visualizzazione del segnale sarà corretta,

4. TASC

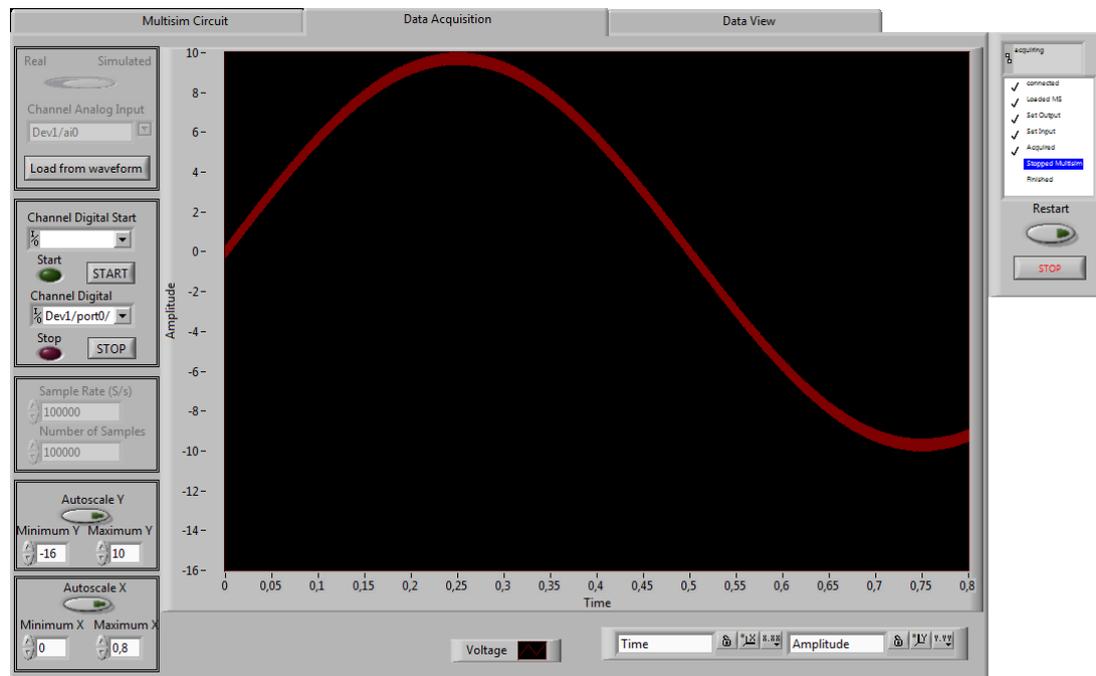


Figura 4.2: Sezione Acquiring di TASC

TASC aspetterà che venga avviata la memorizzazione del segnale che comincerà nel momento in cui verrà premuto il pulsante *START*, vicino alla luce verde. In questo momento il segnale visualizzato nell'oscilloscopio cambierà colore diventando rosso. I dati acquisiti cominciano ad essere memorizzati dal software fino a che non viene premuto il pulsante di *STOP* vicino alla luce rossa che si accenderà, spegnendo la luce verde. L'acquisizione verrà fatta tramite un campionamento con *Sample Rate* che ha un valore di default di 100kHz. Questo è un parametro fondamentale da scegliere, considerando che il campionamento deve rispettare il teorema di Nyquist e deve essere sufficientemente elevata per poter considerare il rumore ad alta frequenza ed avere una buona risoluzione nel dominio del tempo. Il fenomeno di aliasing fornirebbe indicazioni sbagliate sulla misura del segnale e deve essere considerato con attenzione. Nell'esempio di figura 4.2 il segnale è una sinusoide di periodo di un secondo e viene acquisito 100k molto più veloce. Questo vuol dire che se l'acquisizione dura 1 secondo vengono simulati 100 mila campioni in NI Multisim. Il numero di campioni di dati considerati nella simulazione aumenta notevolmente la complessità temporale dell'algoritmo di simulazione e di conseguenza di tutto il test eseguito con TASC. *Sample Rate* è quindi un parametro molto molto importante per la corretta analisi del circuito. *Number of*

sample indica la dimensione del buffer di memoria in cui vengono memorizzati i dati acquisiti. Nel caso preso in esame *Number of sample* è di 100k campioni, che è lo stesso valore di *Sample Rate*. Ciò comporta che il software trascura i comandi di inizio e fine memorizzazioni ad istanti multipli di 1 secondo. Per un miglior funzionamento di TASC è consigliabile che *Numero of sample* rimanga di ordine inferiore rispetto a *Sample Rate*. I comandi di inizio e fine memorizzazione sono gestibili anche attraverso comandi digitali esterni al software. Infatti TASC è stato pensato per poter acquisire due segnali digitali esterni in ingresso alla scheda di acquisizione. I canali da selezionare sono rispettivamente *Channel digital start* e *Channel digital stop*. Queste opzioni permettono di definire gli istanti precisi di inizio e fine acquisizione. La figura 4.3 rappresenta un esempio di un segnale in

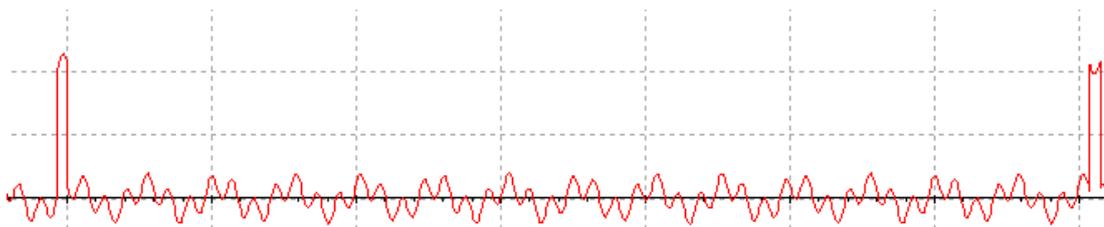


Figura 4.3: Esempio di segnale compreso tra enti rilevanti

cui è utile, anzi indispensabile, utilizzare i comandi digitali per far partire e fermare l'acquisizione. Il segnale compreso tra i due eventi dura circa 14ms. Questo tempo è troppo ridotto per poter essere comandato dall'utente, mentre è possibile rilevare i 2 eventi tramite acquisizione digitale che determineranno poi gli eventi di *START STOP*. Avviata la memorizzazione dei dati acquisiti, i comandi di selezione del canale (setting dell'acquisizione) verranno disabilitati per evitare errori. La scheda di acquisizione utilizzata in laboratorio per gli esperimenti è NI ELVIS che ha una frequenza massima di campionamento di 1,2 MHz e 16 bit di quantizzazione. Il range di ingresso di questa scheda di acquisizione è di 10Volt. Questi dati servono a determinare l'errore massimo di quantizzazione e frequenza del segnale d'ingresso massima

$$Q = \frac{Range}{2^b}$$

$$F_{segnale} < \frac{F_c}{2}$$

Questi sono parametri del campionamento della scheda di acquisizione come discusso nel capitolo 3 e non della visualizzazione del segnale. La risoluzione grafica dipende dalle impostazioni della risoluzione del calcolatore su cui TASC sta effettuando l'analisi e quindi non possono essere specifiche indipendenti. Per questi motivi TASC lascia all'utente la possibilità di impostare la scala e l'offset. I comandi di *autoscale* degli assi permettono di ingrandire o rimpicciolire una porzione dello schermo dell'oscilloscopio. In figura 4.2 è stato impostato un valore minimo dell'asse X di 0 e un massimo di 0.8. Il segnale è a frequenza di 1 Hz e quindi non può essere visualizzato interamente. Con questa configurazione la risoluzione orizzontale sarà migliore. Anche nell'asse delle ampiezze è possibile impostare la scala facendo variare il valore minimo e massimo visualizzato. Queste impostazioni sono state pensate per poter notare piccole variazioni nel dominio del tempo o delle ampiezze se ci fosse la necessità di ingrandire la sezione di un segnale. Si consiglia di utilizzare le impostazioni di *autoscale* visualizzando l'intero segnale e, solo in casi particolari, diminuire la risoluzione per vedere i dettagli di un segnale.

TASC permette anche di non acquisire un segnale, né reale né simulato, ma di utilizzare un segnale precedentemente salvato come waveform. Questa opzione permette di effettuare l'analisi e lo studio del circuito anche in tempi e luoghi diversi da dove e quando è stato acquisito il segnale reale, utilizzandolo come se fosse appena stato acquisito. Il pulsante *Load from waveform* apre una finestra di dialogo che consente di scegliere quale file caricare, se utilizzare il segnale d'ingresso o di uscita e di poter cambiare tutti i valori delle ampiezze amplificando il segnale o variandone la frequenza. Nella directory di installazione di TASC vengono installate librerie contenenti i più frequenti segnali reali periodici come sinusoidi, onde triangolari e quadre. Nella libreria sono presenti anche diversi modelli di rumori reali che possono essere amplificati e aggiunti al segnale periodico.

4.3 Elaborazione dati

La sezione di visualizzazione dati è divisa in due parti esattamente uguali sovrapposte e da una barra degli strumenti sottostante. Questa schermata viene visualizzata automaticamente quando l'utente ha finito l'acquisizione. Il segnale acquisito

e memorizzato viene inserito nel circuito NI Multisim e visualizzato nell'oscilloscopio superiore di questa sezione di TASC assieme al segnale d'uscita. In questo caso l'acquisizione del segnale sinusoidale a frequenza di 1Hz dura esattamente 1,6 secondi anche se nella sezione acquiring ne si visualizzava solo la prima metà. Questa schermata fornisce all'utilizzatore di TASC molte informazioni importanti come la visualizzazioni del segnale d'ingresso e dell'uscita. Dal menu a tendina evidenziato con la lettera B in figura 4.4 si può scegliere se visualizzare l'ingresso o l'uscita o entrambi. Gli oscilloscopi mostrano l'intero segnale acquisito e non una

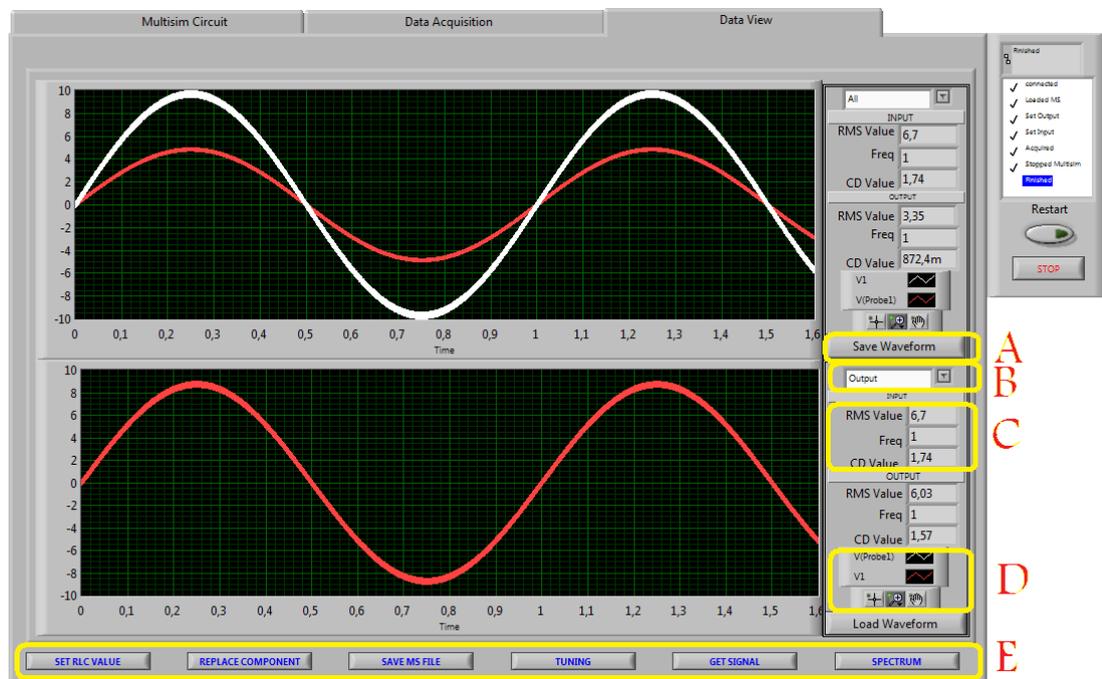


Figura 4.4: Pannello frontale di TASC per l'elaborazione dati

finestratura come solitamente succede negli oscilloscopi tradizionali. I comandi indicizzati con la lettere D servono a permettere all'utilizzatore di spostare, ingrandire o rimpicciolire il segnale all'interno del display a sfondo nero per poter visualizzare una piccola o breve componente del segnale. I riferimenti numerici negli assi dell'oscilloscopio sono dinamici, il che vuol dire che non serve impostare un valore V/Div o un valore di Time/Div e contare le divisioni occupate dal segnale, ma i riferimenti degli assi possono assumere valori adeguati per la sezione quando la parte interessata del segnale viene selezionata e zoomata. Come già detto nel capitolo 3 la risoluzione di questo oscilloscopio dipende dalle caratteristiche della scheda di acquisizione, dai bit di quantizzazione, dalla frequenza di

campionamento, ma anche dalle impostazioni dello schermo del computer che, tutte sommate, definiscono il limite della risoluzione temporale e nell'asse delle ampiezze. Labview offre queste comode e veloci rappresentazioni grafiche come strumenti di analisi.

TASC permette di salvare sul disco fisso i grafici visualizzati nell'oscilloscopio superiore per poi poterli visualizzare su quello inferiore anche in istanti diversi. Queste funzioni sono evidenziate in figura 4.4 con la lettera A. Come visto nel capitolo 4.2, TASC permette di utilizzare come segnale d'ingresso un waveform caricandolo dal disco fisso, il che consente di acquisire un segnale, salvarlo e poterlo riutilizzare per la progettazione ed analisi in luoghi e tempi diversi da quello di origine. Si noti che se si desidera solo visualizzare un grafico precedentemente salvato è possibile cliccare sul pulsante *Load Waveform* e tale segnale verrà visualizzato sull'oscilloscopio inferiore, consentendo a un primo paragone tra il segnale precedentemente utilizzato e visualizzato sullo strumento superiore e quello appena caricato nello strumento inferiore. Il software è stato pensato per poter fornire alcune indicazioni numeriche sui segnali d'ingresso e di uscita, come la frequenza, se si tratta di segnali periodici, il valore efficace o ancora il valore medio. Questi tre parametri caratteristici dei segnali reali possono dare indicazioni di amplificazioni, distorsioni, effetti rumorosi ecc. La sezione contrassegnata con la lettera E è composta da sei strumenti molto importanti per la progettazione ad hoc e sono i seguenti:

- *SET RLC VALUE*: Premendo questo pulsante vengono eseguite le istruzioni di uno stato che ha come centro logico Multisim Set RLC Value. La pressione del pulsante porta TASC a visualizzare di nuovo la prima sezione Multisim Circuit, abilitando però la parte di New Vaule e il pulsante Set New Values come mostrato in figura 4.5. A questo punto diventa possibile variare tutti i componenti RLC del circuito utilizzando esattamente le stesse condizioni di setting e lo stesso segnale, ma con i nuovi valori dei componenti. Per fare ciò è sufficiente cambiare i valori e premere il pulsante Set New Values. Nell'esempio mostrato in figura 4.5 si vuole testare un circuito amplificatore a BJT moltiplicando il valore del primo condensatore della lista di un fattore 10. Una volta premuto il pulsante Set New Values TASC torna nella Data View e visualizza sull'oscilloscopio inferiore il segnale d'ingresso e di uscita del circuito con le nuove impostazioni come

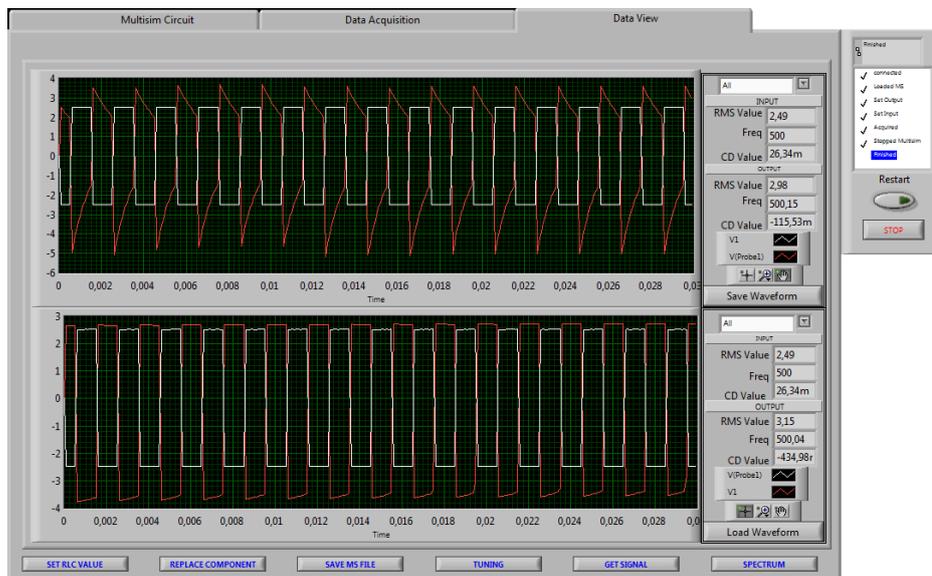


Figura 4.6: Visualizzazione dei nuovi dati con i valori dei componenti cambiati

loscopio inferiore il segnale prodotto dal circuito con il nuovo componente.

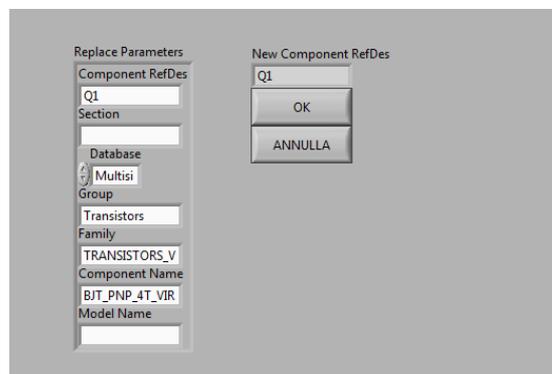


Figura 4.7: Finestra di dialogo per sostituire un componente del circuito

- *SAVE MS FILE*: Il tasto *Save MS File* mostra all'utente la piccola finestra di dialogo di figura 4.8 che consente di salvare o sovrascrivere il file Multisim modificato cambiando i componenti RLC o sostituendo gli altri componenti. Dopo aver modificato il circuito fino ad ottenere il segnale d'uscita desiderato sarà possibile salvare il progetto.
- *TUNING*: Il pulsante *Tuning* fa entrare TASC in uno stato dall'architettura molto sofisticata ma che esegue istruzioni molto semplici ed utili. Tuning

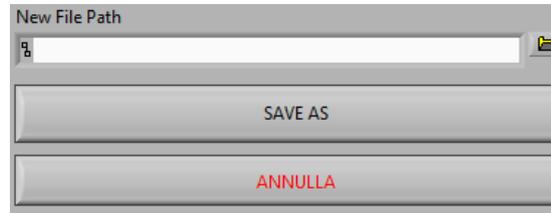


Figura 4.8: Finestra di dialogo per salvataggio del file Multisim

permette di trovare la giusta configurazione del circuito per ottenere un segnale d'uscita che abbia determinate proprietà richieste dall'utilizzatore. Nell'esempio in figura 4.9 è richiesto che il segnale d'uscita abbia un valore RMS maggiore di $0.3V$, che vengano effettuate 10 prove facendo variare il valore di $R1$ da $10k\Omega$ a $3k\Omega$. Dopo aver premuto OK il software eseguirà 10 simulazioni partendo con $R1 = 10k\Omega$ a intervalli regolari fino a simulare il circuito la decima volta con $R1 = 3k\Omega$. Ad ogni ciclo viene calcolato il valore RMS del segnale e confrontato con quello richiesto. Se la condizione è verificata il ciclo termina e TASC visualizzerà nell'oscilloscopio inferiore il segnale d'uscita, altrimenti eseguirà tante simulazioni quante sono richieste e poi metterà in grafico l'ultimo segnale d'uscita. Durante le iterazioni vengono visualizzati tutti i segnali e l'utente può fermare il ciclo quando preme il pulsante STOP. A questo punto si chiuderà l'interfaccia e verrà visualizzato l'ultimo segnale generato. Dopo il tuning è possibile premere il tasto *Save MS File* e salvare il circuito simulato, che ora ha il valore RMS desiderato. Così facendo, TASC calcolerà il valore del componente che si vuole variare, tale da verificare la condizione richiesta. Il tuning effettua molte simulazioni, quindi ha una complessità temporale è proporzionale al tempo impiegato per una singola simulazione moltiplicato per il numero di iterazioni effettuate.

- *GET SIGNAL*: Questa sezione è utile in quanto il progettista può generare, o salvare su file, un segnale reale uguale a quello generato dall'uscita del circuito. Spesso è molto utile poter utilizzare il segnale reale del circuito simulato perché rende possibile usare TASC come ponte tra due hardware reali. Esempi di questo tipo sono evidenziati nel capitolo 5.1 e 5.2. TASC può comportarsi come un vero e proprio circuito reale: può utilizzare come ingresso un segnale reale e fornire un'uscita che a sua volta può essere utiliz-

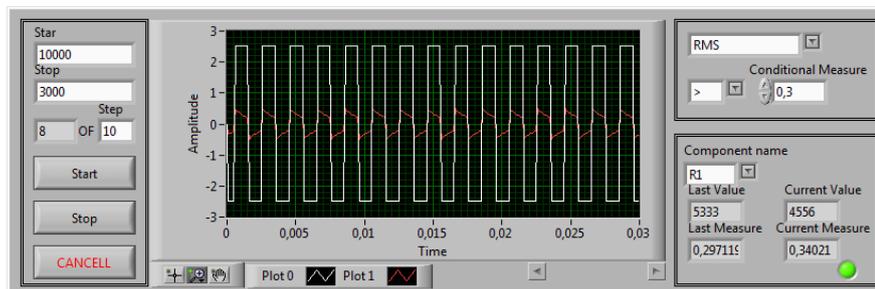


Figura 4.9: Interfaccia utente del tuning del circuito

zata da un hardware esterno, collaudando il circuito simulato in condizioni reali.

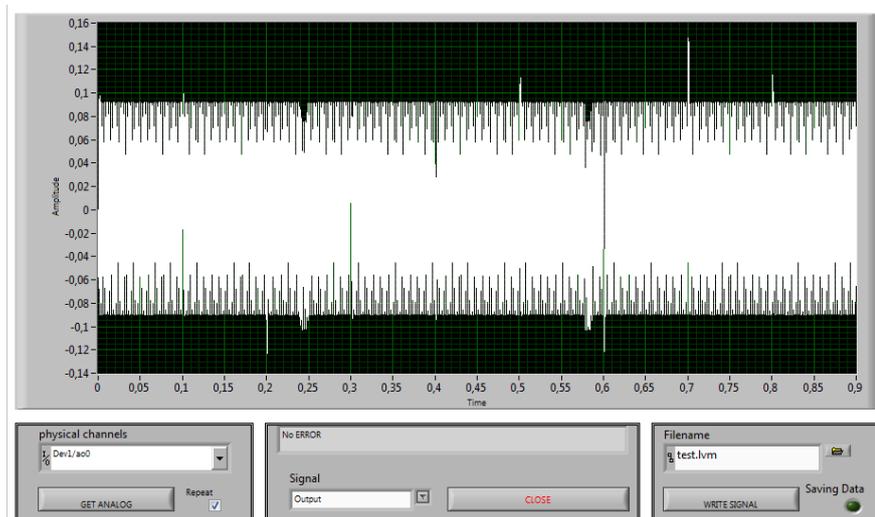


Figura 4.10: Get Analog Output

- SPECTRUM:** TASC è un software di esempio per l'utilizzo di Multisim Connectivity Toolkit e come tale è stato pensato per poter studiare il circuito anche nel dominio delle frequenze sia dell'ingresso che dell'uscita. In figura 4.11 è mostrato l'andamento in frequenza di un filtro con segnale d'ingresso quadro, di frequenza 500 Hz, campionato a 10 kHz. Nella stessa figura si nota la differenza tra le componenti spettrali del filtro con 2 valori diversi di un condensatore. Anche in questo caso è possibile confrontare il segnale del circuito originale con quello del circuito modificato, sostituendo un componente o cambiandone il valore.

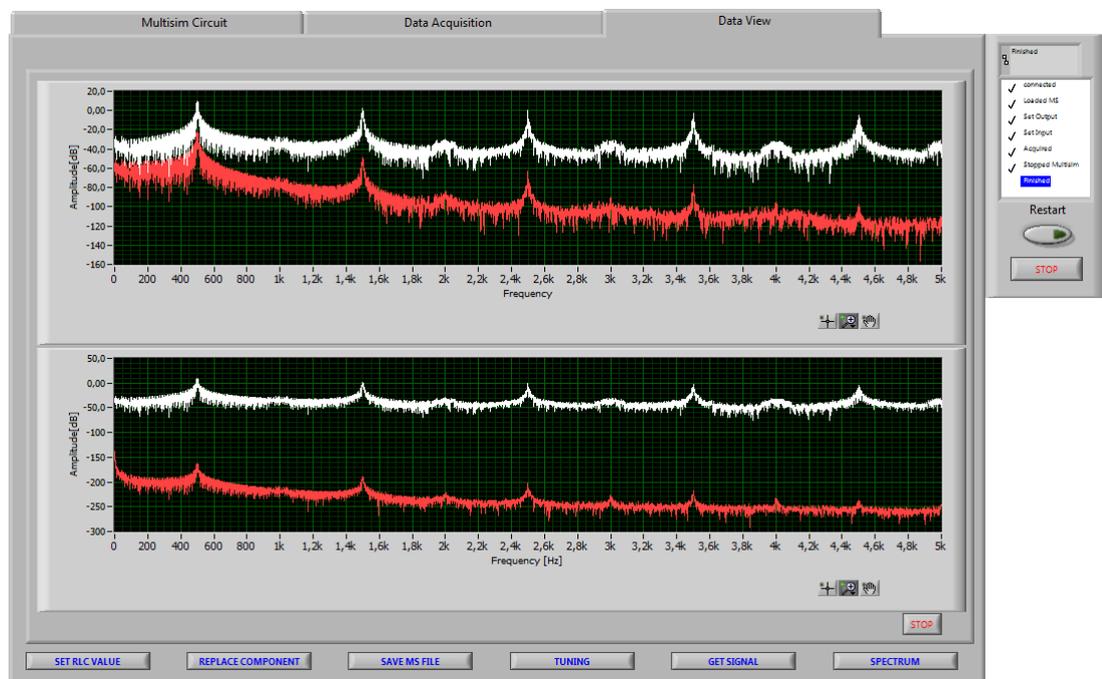


Figura 4.11: Analizzatore di spettro

Capitolo 5

Misure

L'integrazione tra simulazione numerica e dati reali si è dimostrata fondamentale in fase di sviluppo e progettazione di alcuni esempi creati in laboratorio. Nelle seguenti sezioni verrà eseguito passo passo il percorso fatto per raggiungere gli obiettivi prefissati utilizzando NI Multisim, Labview e NI ELVIS.

5.1 Ventola raffreddamento per CPU

Si dispone di una ventola di raffreddamento utilizzata per abbassare la temperatura dei processori Pentium IV. Questa ventola dispone di due collegamenti di alimentazione e un terzo filo che fornisce un segnale elettrico periodico di periodo proporzionale alla velocità di rotazione della ventola. Per lo studio della ventola è stato utilizzato il generatore power supply di NI ELVIS. È stata impostata una tensione di alimentazione pari a $6V$ e la ventola a fornito un segnale mostrato in figura 5.1.

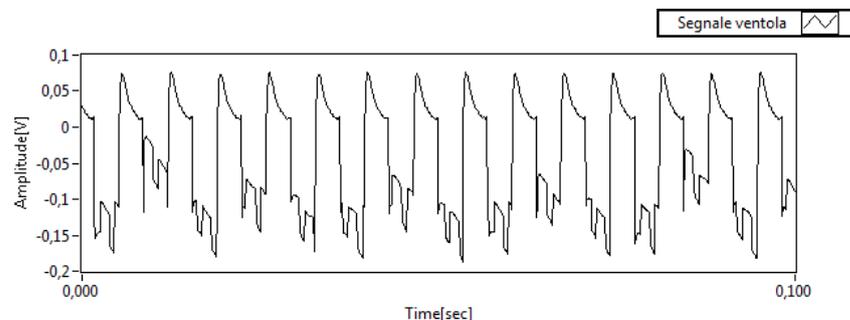


Figura 5.1: segnale fornito dalla ventola sottoposta ad una caduta di tensione di $6V$

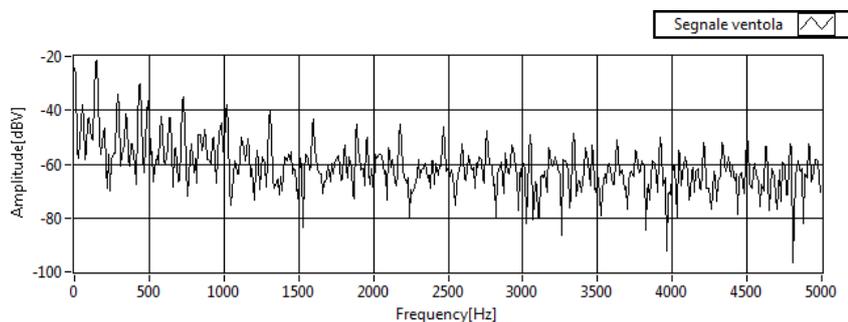


Figura 5.2: spettro del segnale fornito dalla ventola sottoposta ad una caduta di tensione di 6V

L'acquisizione di ELVIS permette in range del segnale d'ingresso di 20 V e lo quantizza in 65536 livelli. Ogni livello ha un'ampiezza pari al rapporto dei due, cioè $q = 0.000305V$. Il segnale in figura 5.1 ha ampiezze nell'ordine dei decimi di Volt. Una quantizzazione con lo stesso range ma a 8 bit cambia q in $0.078125V$. Con un quantizzatore ad 8 bit e senza adattamento di scala non sarebbe stato possibile vedere variazioni inferiori a 78mV. ELVIS utilizza 16 bit e un sistema di adattamento di scala che permette di adattare il segnale per poter utilizzare sempre di tutti i bit a disposizione. Questo permette di poter acquisire correttamente e con buona approssimazione anche segnali piccoli come in questo caso. Le tre pale della ventola generano uno particolare impulso ogni volta che passano per un punto fisso. Questo vuol dire che la frequenza del segnale elettrico di figura 5.1 è esattamente tre volte maggiore della velocità angolare della ventola.

Si supponga di voler realizzare un circuito che conti i giri al secondo della ventola. La relazione tra caduta di tensione e giri/minuto non è lineare come si è notato sperimentalmente. Il segnale di figura 5.1 può essere utile proprio a questo scopo. Nelle figure 5.1 e 5.2 si può notare che il segnale utile è irregolare e di ampiezza nell'ordine dei decimi di Volt che può non essere sufficientemente elevato per distinguerlo dal rumore. L'idea è di utilizzare un circuito a display che conti il numero di intersezioni del segnale sull'asse delle ascisse. Il contatore di impulsi del capitolo 1.2.4 fa proprio al caso nostro, ma bisogna prima pulire il segnale senza variarne la frequenza. Una possibile soluzione è di utilizzare dei circuiti tipo 1.2.1 e 1.2.2 del primo capitolo. Con Multisim Wizard si è realizzato un circuito che contiene un amplificatore operazionale e un filtro passa basso. Senza fare calcoli impegnativi su frequenza di taglio e attenuazioni di un filtro di ordine 10 è stato costruito un circuito Multisim contenente filtro e amplificatore. Usan-

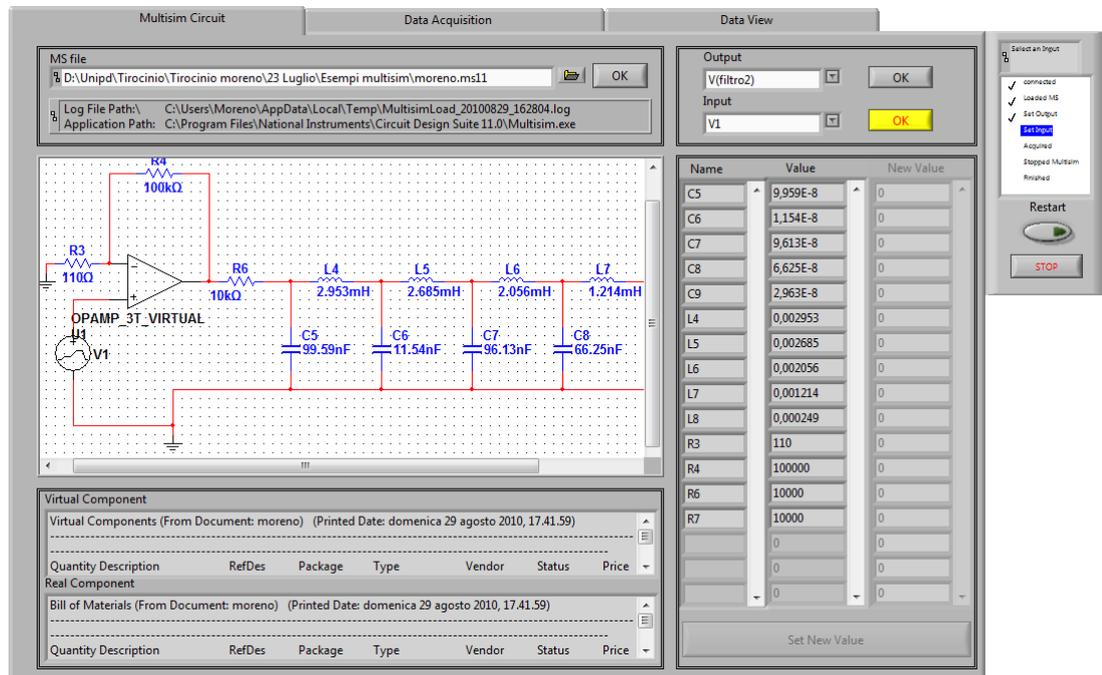


Figura 5.3: Collaudo dal circuito proposto da Multisim Wizard

do TASC è stato collaudato il circuito con il vero segnale della ventola di figura 5.1. Il risultato è mostrato in figura 5.4 e permette di poter trarre senza nessuna fatica di conti in fase di progettazione, una prima serie di osservazioni sul circuito in esame. Si ricorda che la progettazione del circuito è stata fatta interamente da Multisim Wizard. Osservando l'oscilloscopio superiore di figura 5.4 si nota che il segnale di uscita ha un'ampiezza attorno ai 5V e che questa caratteristica può soddisfare una nostra specifica. Ora si tratta di avere dei fronti di salita più veloci, avvicinandosi il più possibile ad un'onda quadrata. La prima cosa da fare per risolvere questo problema è di provare a ridurre l'ordine di grandezza di condensatori presenti per accorciare il tempo di carica e di scarica. Per fare questo si preme il pulsante SET RCL VALUES e si abbassano tutti i condensatori ad un valore 10 volte più piccolo. Si preme SET NEW VALUES. TASC elabora il segnale attraverso il circuito con i condensatore di valore più piccolo rispetto ai precedenti ed elabora nuovamente il risultato mostrato in figura 5.4 nell'oscilloscopio inferiore. A questo punto si osserva che il segnale di uscita in rosso è del tutto simile ad un'onda quadrata, che le frequenze di ingresso e di uscita sono le stesse medesime e che l'ampiezza è adeguatamente alta. Il circuito Multisim viene salvato.

5. MISURE

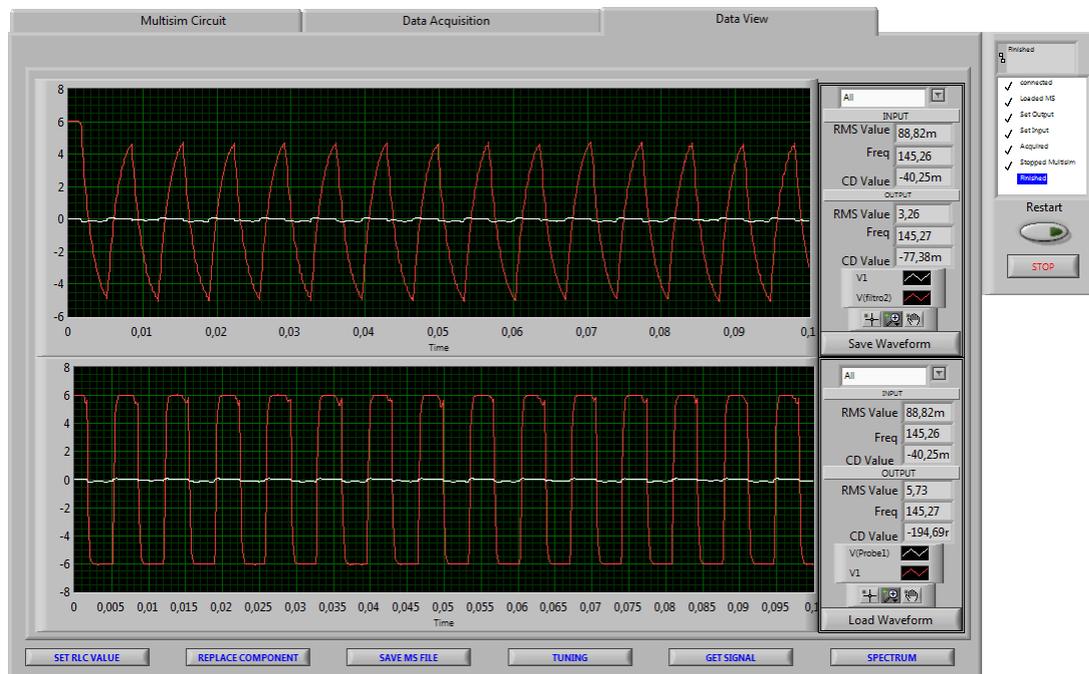


Figura 5.4: Segnale prodotto dal circuito dopo le modifiche

Il circuito contatore di eventi a tre display del paragrafo 1.2.4 è stato fisicamente costruito sulla beadBord di NI ELVIS. Il segnale di uscita del circuito filtro+ amplificatore è stato generato attraverso ELVIS è inserito nel circuito reale contatore. Per un segnale della ventola di durata di secondo, l'indicatore a display mostrava il valore numerico di 146 fronti d'onda positivi. Nell'indicazione di frequenza di TASC di figura 5.3 è mostrata una frequenza portante di 145.27Hz. Questo numero è molto più preciso di quello visualizzabile sul circuito contatore di eventi a display per diversi motivi, ma l'obbiettivo che si voleva raggiungere era di costruire un circuito che permetta ottenere un'indicazioni sul numero di giri al minuto della ventola a partire da un piccolo segnale particolare di figura 5.1.

$$\text{Giri al minuto} = \text{secondi in un minuto} \frac{\text{frequenza}}{\text{numerodipale}} = 60 \frac{146}{3} = 2029$$

Applicando una tensione di 6V il motore compie 2029 giri al minuto.

L'errore commesso con questa misura è al massimo di un Hz che equivale a 20 giro al minuto. Dal tipo di utilizzo che si fa di questa misura l'errore è trascurabile o meno, ma questo esempio voleva dimostrare che il metodo di integrazione nu-

merica di NI Multisim e l'acquisizione di dati reali di NI Labview più essere molo comoda per la progettazione di un qualsiasi circuito. È stato possibile progettare un circuito elettronico che assicuri al progettista il corretto funzionamento anche nella realtà avendo a disposizione la ventola che genera il segnale di interesse e il contatore di aventi, prima solo simulato e poi realizzato fisicamente. È stata fatta una brevissima analisi del segnale reale, si è abbozzato un circuito che fornisca come uscita il segnale desiderato ed sono stati fatto poche modifiche fino ad ottenere il segnale desiderato.

5.2 Trasmissione segnale binario

Con questa esperienza si può intuire l'utilità di utilizzo di Multisim Connectivity Toolkit. I circuiti che verranno presi in considerazione sono un circuito di modulazione AM, simile al circuito 1.5, ed un circuito demodulatore. Verrà testato un sistema completo di trasmissione di un segnale binario. Il segnale reale acquisito è una parola binaria a 24 bit con durata totale pari a 40ms. Questo segnale è stato generato da una scheda per lo studio dei sistemi di modulazione e demodulazione montata su ELVIS al posto della scheda base con breadboard. Il segnale verrà poi moltiplicato per una sinusoida a frequenza di 5kHz attraverso il circuito di modulazione di figura 5.5 a sinistra. Il modulato verrà generato da ELVIS e trasmesso attraverso tre diversi canali per poter osservare i comportamenti diversi del circuito demodulatore 5.5 a destra. Il segnale digitale acquisito

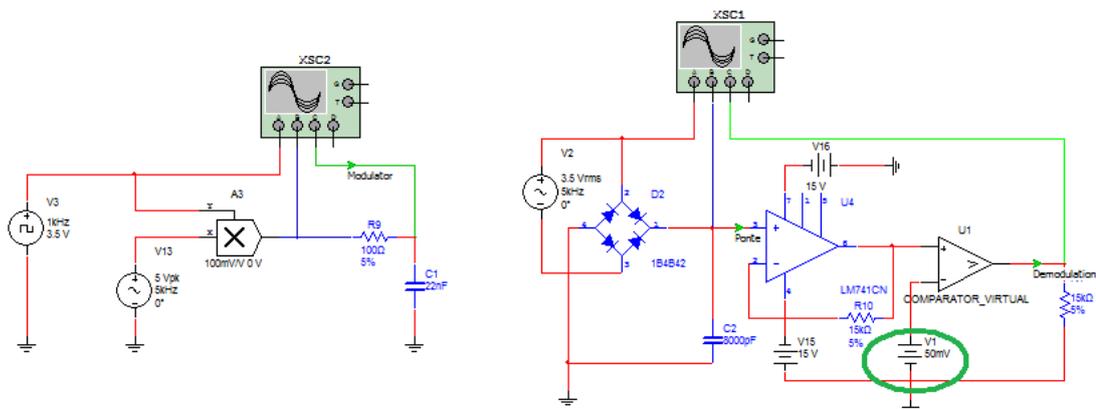


Figura 5.5: Circuiti Multisim di modulazione e demodulazione AM

è campionato a 100kHz, quindi il segnale completo è composto da 4000 campioni.

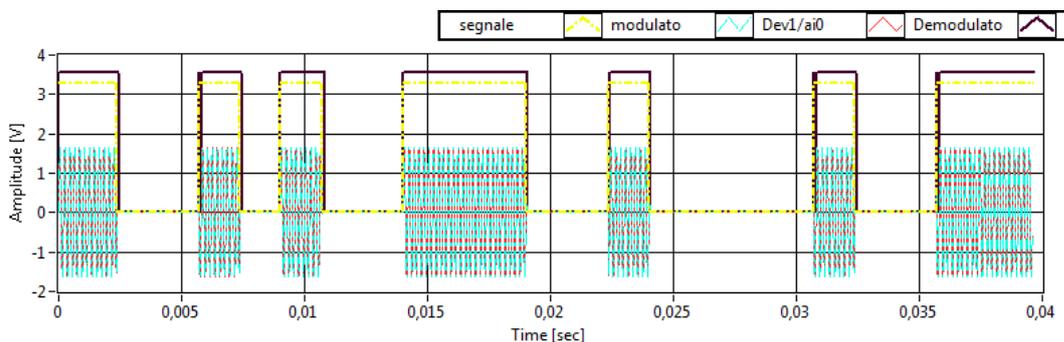


Figura 5.6: Segnale trasmesso (giallo); Segnale modulato e trasmesso (azzurro); Segnale ricevuto dal canale ideale (rosso); Segnale demodulato (viola)

Nel primo test Labview avvia la simulazione del circuito modulatore utilizzando come sorgente il segnale digitale. Il segnale modulato verrà trasferito al circuito demodulatore come variabile Labview. In figura 5.6 si può vedere l'intero percorso del segnale nelle sue diverse fasi: il segnale sorgente in giallo; il segnale modulato in azzurro tratteggiato, il segnale in ingresso al demodulatore in rosso e il segnale ricostruito dal demodulatore in viola. Nel circuito demodulatore il segnale modulato e trasmesso attraverso il canale, viene raddrizzato, fatto passare per un buffer di protezione e comparato poi con una tensione di riferimento a 50 mV (evidenziato in verde) per la ricostruzione esatta del sorgente. Si è scelto di non utilizzare come riferimento la massa del circuito, ma una tensione non nulla per trascurare l'eventuale rumore. Il demodulatore è progettato per poter funzionare correttamente con un rumore di ampiezza massima di 50 mV. In figura 5.6 si può vedere il corretto funzionamento dei circuiti utilizzando un canale ideale di trasmissione.

Il secondo test eseguito verifica il comportamento del demodulatore utilizzando un sottile filo di rame di diametro 1.5mm di lunghezza di 10 cm come canale di trasmissione. In questo caso il segnale modulato viene generato, trasmesso e ricevuto da ELVIS attraverso il canale reale. Nonostante il canale non sia più ideale non si notano differenze sostanziali con il grafico di figura 5.6. Il segnale in viola continua ad essere uguale al segnale in giallo. Ciò vuol dire che il rumore introdotto nel sistema attraverso il canale è inferiore a 50 mV non interferisce nella ricostruzione del segnale. Il terzo test è stato effettuato utilizzando la stessa configurazione dei circuiti degli esempi precedenti e utilizzando un cavo della stessa sezione del precedente ma di lunghezza di 200m. Il filo di rame non è

schermato ed è stato appositamente fatto passare vicino a fonti di rumore come alimentatori e altri strumenti da laboratorio. Il risultato di figura 5.7 mostra che il rumore introdotto dal nuovo canale di trasmissione disturba notevolmente la demodulazione.

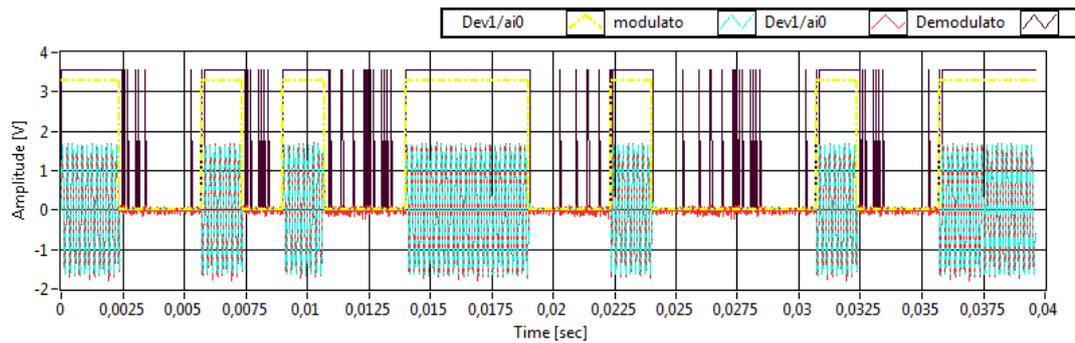


Figura 5.7: Segnale trasmesso (giallo); Segnale modulato e trasmesso (azzurro); Segnale ricevuto attraverso il filo di 200m (rosso); Segnale demodulato (viola)

È facile notare che il segnale ricevuto non è nullo nei punti in cui il segnale binario è a livello logico basso. Si può anche intuire ad occhio nudo che probabilmente, il rumore ha dei picchi maggiori di 50mV. Viene ripetuto il test impostando il valore di V1 a 100mV. Il risultato migliora, ma il rumore supera ancora il valore di soglia del comparatore. Viene effettuato un terzo test raddoppiando ancora il valore di V1. Il risultato è soddisfacente ed è mostrato in figura 5.8.

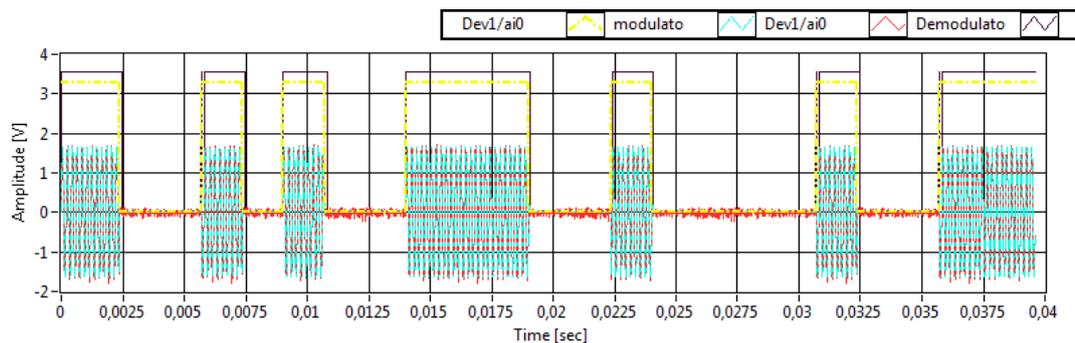


Figura 5.8: Segnale trasmesso (giallo); Segnale modulato e trasmesso (azzurro); Segnale ricevuto attraverso il filo di 200m (rosso); Segnale demodulato corretto (viola)

Si può concludere che è necessario un valore di V_1 pari a 200mV per poter trasmettere il segnale digitale di interesse attraverso le particolari condizioni a cui è sottoposto il canale di trasmissioni. Non si sarebbe potuti arrivare a questa conclusione se non si fosse collaudato l'intero sistema nelle condizioni reali. In fase di progettazione era stato costruito il demodulatore per evitare disturbi stimati fino a 50mV. Il sistema non commette errori se il canale è ideale o se è un filo corto, ma non può essere utilizzato con canale soggetto a forte.

Questo esempio mostra che esiste una notevole differenza tra condizioni reali e condizioni ideali di uno o più circuiti simulati. Lo schema Labview di figura 5.9 mostra il software che permette di collaudare sia il circuito modulatore, sia il demodulatore. Il segnale binario precedentemente acquisito e salvato viene caricato e utilizzato come segnale sorgente che entra nel modulatore. Come prevede l'architettura di Multisim Connectivity Toolkit, viene riservato il segnale d'ingresso e il segnale d'uscita. La simulazione viene fatta durare 40ms utilizzando 4000 campioni a 100k campioni al secondo. Questa scelta è stata fatta per uniformare il segnale d'ingresso con la simulazione. Le stesse impostazioni vengono eseguite per la simulazione del demodulatore. Il VI di centro figura è uno strumento che genera il segnale modulato e contemporaneamente lo acquisisce. Nel primo test il VI di interesse è stato cortocircuitato per rendere il canale virtuale, mentre nel secondo e terzo test il VI è quello che permette di verificare gli effetti dei disturbi che vengono introdotti dal rumore. Per lo sviluppo di questo sistema è stato utilizzato ELVIS sia come generatore di parola binaria, sia per la generazione del segnale binario, sia per acquisire il segnale all'uscita del demodulatore. Il campionamento, la generazione e la frequenza di sincronizzazione di Multisim sono impostati a 100kHz per poter rispettare il teorema di Nyquist ed è sufficientemente elevata per avere un buon campionamento del segnale modulato a 5kHz. Le caratteristiche di ELVIS permettono il corretto funzionamento del circuito di figura 5.9, ma non sarebbe stato adatto se si volesse utilizzare un segnale portante di 1MHz. ELVIS non sarebbe stato in grado di acquisire a frequenza maggiore di 2MHz e quindi la ricostruzione del segnale sarebbe stata non corretta. Lo stesso tipo di osservazioni si possono fare nel dominio delle ampiezze. Lo schema in figura 5.9 mostra anche che i VI di Multisim Connect e Multisim Disconnect sono singoli, ed utilizzato per aprire entrambi i circuiti Multisim. Ovviamente questa struttura richiede due simulazioni differenti di due circuiti differenti, per ognuno

dei quali si richiede di riservare un ingresso, raccogliere i dato in uscita dopo aver avviato la simulazione. Si noti che vengono effettuate due differenti simulazione e che questa strategia non è quella preferibile in termini di complessità temporale.

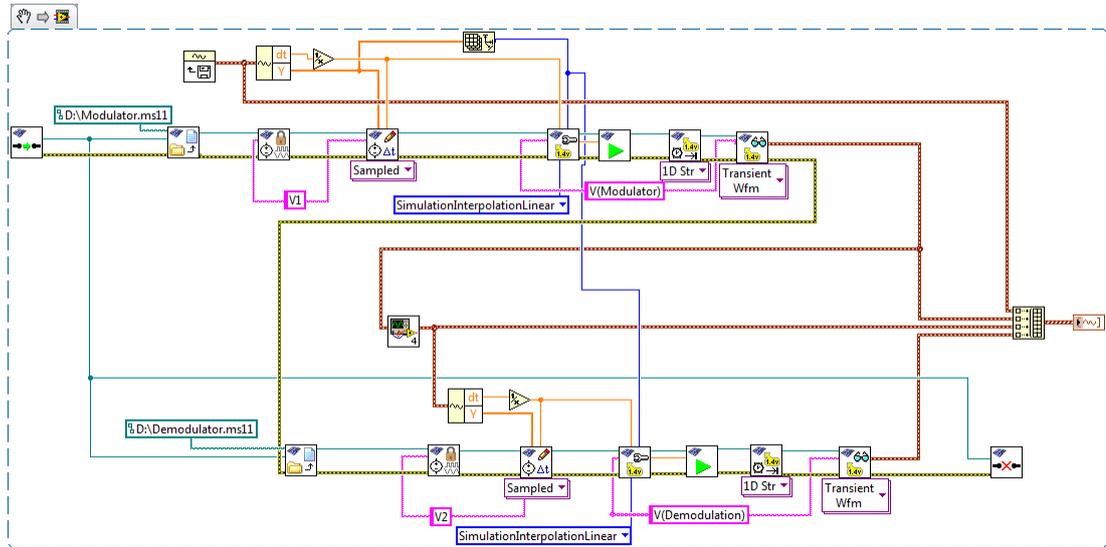


Figura 5.9: Schema Labview per il collaudo della modulazione attraverso Multisim Connectivity Toolkit

5. *MISURE*

Conclusioni

Nel capitolo 3 è stato discusso il campionamento e la quantizzazione e le caratteristiche di ELVIS per poter verificare il corretto funzionamento di TASC. Nel capitolo 5 si sono visti due esempi in cui la frequenza di campionamento e la quantizzazione si sono dimostrati parametri hardware fondamentali. Dipendentemente dalla struttura degli ADC interni anche il numero elevato di bit di quantizzazione può rallentare il processo di conversione e quindi porre un limite superiore alla frequenza a è possibile campionare. Campionare ad alta frequenza non è quindi sempre possibile, anzi a volte non è neppure conveniente. Avere una buona risoluzione del dominio del tempo comporta una bassa risoluzione nel dominio delle frequenze. TASC prevede entrambe le analisi e quindi sta all'utente scegliere la frequenza di campionamento opportuna in base al tipo di analisi che si vuole fare per il collaudo del circuito. Nell'esempio 5.1 il segnale di interesse è relativamente lento, per cui i vantaggi dovuti ad un campionamento veloce non compensa la bassa risoluzione dell'analizzatore di spettro. Il circuito dell'esempio 5.1 è un filtro di decimo grado e richiede anche un notevole tempo necessario alla simulazione. La durata della simulazione è proporzionale al numero di punti che compongono il segnale d'ingresso. Campionare il segnale della ventola a 100kHz comporta una durata della simulazione dieci volte più grande di quella che impiegherebbe Multisim a simulare il circuito con un segnale campionato a 10kHz. Essendo un filtro si preferisce l'analisi nel dominio delle frequenze per cui è opportuno che il campionamento non sia fatto ad alta frequenza. Nell'esempio 5.2 si preferisce campionare ad alta frequenza per poter avere più informazioni possibili del comportamento del demodulatore nel dominio temporale. Per questi circuiti è più importante avere una buona risoluzione temporale per poter rilevare i fenomeni evidenziato in figura 5.7.

In entrambi i casi è importante verificare le caratteristiche della scheda di acquisizione che si sta utilizzando per non incorrere in errori o non rischiare di

5. MISURE

visualizzare segnali non corretti. Sta quindi all'utente la giusta valutazione del campionamento, dei risultati e della scheda di acquisizione adatta.

In tutti gli esempi presi in considerazione si è fatto riferimento solo a circuiti simulati di cui venivano gestiti un unico generatore e un'unica probe. Multisim Connectivity Toolkit non ha limitazioni per il numero di variabili gestite da Labview. Le considerazioni fatte per un sistema Singol Input Singol Output sono le medesime che si possono fare per sistemi Multiple Input Multiple Output (o loro combinazioni).

Bibliografia

LabVIEW Corso Base I sulla versione 8.6 [2008]

LabVIEW Corso Base II sulla versione 8.6 [2008]

LabVIEW: Acquisizione dati e condizionamento dei segnali sulla versione 8.6 [2008]

<http://www.patrucco.it/educad/17/multisim.html>

<http://sine.ni.com/nips/cds/view/p/lang/it/nid/13137>

<http://www.ni.com/multisim/i/>

<http://www.ni.com/labview/i/>

<http://zone.ni.com/devzone/cda/tut/p/id/7785>