

POLITECNICO DI TORINO

Facoltà di Ingegneria

Corso di Laurea in Ingegneria Elettrica Magistrale

Tesi di Laurea

**Procedura automatica per il calcolo
delle correnti di cortocircuito e dei flussi
di potenza nella rete tranviaria torinese**



Relatori:

Prof. Ing. Enrico Pons

Dott. Ing. Pietro Colella

Dott. Ing. Roberto Rizzoli - Infra.To

Candidato:

LORENZO BERTOLONE CITIN

Luglio 2018

*Dedicata ai miei genitori Daniela e Claudio,
alle mie sorelle Francesca e Sathya,
ai nonni Melina, Salvatore e Luigi,
ai miei cugini, zii, e Alessandra.*

Grazie per aver sempre creduto in me.

Indice

1	Introduzione	1
2	Sistema Tranviario Torinese	2
2.1	La Rete Elettrica	2
2.2	Le Motrici	3
3	Interfaccia al Simulatore di rete tranviaria software RHS di Genova	6
3.1	Consultazione/editazione della rete	7
3.2	Importazione dei dati ed assegnazione della zona	8
3.3	Calcolo e visualizzazione dei risultati	8
4	Progetto 'Augusta Tauringrid'	11
4.1	Esportazione da ambiente Autocad	12
4.2	Configurazione plugin ed importazione in ambiente QGIS	18
4.3	Elaborazione dati in ambiente QGIS	24
4.4	Esportazione da ambiente QGIS	41
4.5	Importazione in ambiente MATLAB	43
4.6	Elaborazione dati in ambiente MATLAB	51
4.7	Visualizzazione dei Risultati su QGIS	53
5	Simulazione di 'Augusta Tauringrid'	65
5.1	Scenario 1: Rete elettrica tranviaria zona 40 in condizioni ordinarie	67
5.1.1	Analisi dei Risultati	67
5.2	Scenario 2: Rete elettrica tranviaria zona Politecnico di Torino in condizioni di guasto a terra	69
5.2.1	Analisi dei risultati	69
5.2.2	Verifica dei risultati	70
5.3	Scenario 3: Rete elettrica tranviaria zona circostante 30 Politecnico di Torino in condizioni di guasto a terra	72
5.3.1	Analisi dei risultati	72
5.3.2	Verifica dei risultati	73

5.4	Scenario 4: Rete elettrica tranviaria in condizioni di guasto a terra nella zona 40 alimentata da due sottostazioni elettriche	75
5.4.1	Analisi dei risultati	75
5.4.2	Verifica dei risultati	76
6	Conclusioni	78
	Indice delle figure	84
	Appendice	85
A	Codice Matlab Augusta Tauringrid	85
B	Licenze Materiale Multimediale	138
	Ringraziamenti	I

Capitolo 1

Introduzione

La rete tranviaria di Torino possiede 210 km di binari singoli, 500 km di cavi sotterranei, 49 sottozone elettriche alimentate da circa 20 cabine di conversione e circa 700 scambi totali. Ogni giorno centinaia di passeggeri usufruiscono del servizio di trasporto pubblico attraverso una o più delle 550 fermate dislocate su tutto il territorio. La complessa magliatura della rete permette non solo di ottenere normali percorsi di linea, ma anche collegamenti con depositi tranviari e deviazioni su tutta la rete ove necessario [1].

Infra.To è una società di progettazione e di realizzazione delle infrastrutture di proprietà della città di Torino nata nell'ottobre del 2010 a seguito della scissione di ramo d'azienda di Gruppo Trasporti Torino (**G.T.T.**) per separare la gestione dei servizi di trasporto pubblico locale dalla gestione delle infrastrutture [2]. Questa tesi è stata sviluppata nell'ambito di un progetto di ricerca stipulato tra Infra.To e il Dipartimento Energia del Politecnico di Torino. *L'obiettivo della tesi* è realizzare una semplice interfaccia grafica, che permetta di risolvere la rete e consultare dati in differenti scenari e configurazioni di sistema per la progettazione, manutenzione, disegno ed aggiornamento della rete elettrica tranviaria. Con la supervisione del personale Infra.To, tramite incontri programmati e dei relatori della tesi, si è deciso di sfruttare l'ambiente di programmazione MATLAB per lo sviluppo dei calcoli e l'ambiente QGIS per l'interfaccia di visualizzazione grafica. La tesi fornisce brevi cenni sul sistema tranviario torinese al capitolo secondo, con l'intenzione di presentare la configurazione della rete elettrica e le motrici utilizzate per far comprendere al lettore le condizioni al contorno su cui verte la tesi e definire i vincoli fisici del sistema. Successivamente, nel capitolo terzo viene sviluppata un'analisi del software utilizzato da Infra.To allo scopo di analizzarne difetti e pregi. Si presenta infine la procedura di realizzazione ed il manuale di utilizzo del nuovo software implementato che prende il nome di **Augusta Tauringrid**.

Capitolo 2

Sistema Tranviario Torinese

2.1 La Rete Elettrica

La rete elettrica tranviaria è suddivisa in zone e ciascuna zona è alimentata da una sola sottostazione elettrica. Dalla linea in media tensione si giunge alle cabine di conversione dove si passa da un sistema in corrente alternata ad un sistema in corrente continua raddrizzata a bassa tensione pari a 600 V. Dalle cabine di conversione l'energia elettrica viene trasferita lungo cavi sotterranei in cassette di alimentazione dalle quali attraverso un complesso sistema di cavi positivi, negativi, di alimentazione e barrature in rame di collegamento si giunge alle linee di contatto costituenti la rete aerea. In ogni sottostazione elettrica ci sono uno o due trasformatori esafase dove a valle di ognuno c'è un solo gruppo di conversione. Ogni cabina di conversione ha delle partenze ognuna delle quali è protetta da un interruttore extrarapido. La rete aerea è suddivisa in zone tramite divisori di zona che sono costituiti da circa 70 cm di materiale isolante. Qualsiasi motrice necessita di una rete di alimentazione elettrica dalla quale poter assorbire corrente per il corretto funzionamento. Per far ciò il tram sfrutta un pantografo che struscia su di una linea di contatto in rame sospesa sopra ad ogni binario ad un'altezza media compresa tra i 4 e i 6 m di altezza sostenuta da un sistema di tiranti composti da funi in materiale isolante. La linea di contatto non è sospesa al centro delle due rotaie del binario ma ha un andamento di poligonazione a *zig zag* affinché non vi sia disuniformità di usura dello strisciante in grafite del pantografo [1]. La corrente assorbita dalla motrice ritorna al generatore attraverso le ruote sulle rotaie ed attraverso cavi negativi e cavi sotterranei chiudono il circuito di alimentazione. Una motrice che transita sotto il divisore deve disinserire la trazione elettrica affinché non sussista la formazione di archi elettrici che producano danni al divisore ed al pantografo. Trattando nello specifico l'argomento dei binari, è definito interbinario la distanza tra le sagome di due mezzi tranviari paralleli che procedono nello stesso senso oppure in quello opposto che deve essere sempre necessariamente superiore ai 40 cm e se non rispetta tale requisito necessita di segnalazione al manovratore tramite segnaletica apposita per dare

la precedenza al mezzo con senso di marcia opposto. Sulla rete tranviaria torinese sono presenti due tipologie di scambio elettrici: la cassa di manovra tradizionale e la cassa di manovra moderna. Entrambe le tipologie sono comandate elettricamente e possono essere azionate manualmente azionando un'apposita leva usualmente a bordo della motrice, da infilare in apposita feritoia. La *cassa di manovra tradizionale* sfrutta un attuatore elettrico con comando a radiofrequenza installata al di fuori della cassa che provoca lo spostamento dell'elemento mobile che determina la diramazione o la confluenza delle rotaie. Il comando a radiofrequenza viene attivato quando la motrice si trova a circa 23 m prima dello scambio e nella rete tranviaria di Torino sono circa presenti 240 scambi di questo tipo. La *cassa di manovra moderna* di dimensioni maggiori invece contiene azionamenti elettrici o elettroidraulici di ultima generazione per l'azionamento a distanza. Gli scambi a molla invece consentono l'accesso ad una zona successiva ma impediscono l'uscita dalla medesima zona in contromano e non sono controllabili con comandi a radiofrequenza e necessitano di apposita segnaletica [1].

2.2 Le Motrici

Pur non concentrando l'attenzione della tesi sui mezzi è necessario almeno conoscerne le *caratteristiche principali* e gli *aspetti costruttivi* in maniera tale da comprenderne il funzionamento durante la simulazione della rete. Su tutto il capoluogo piemontese si manovrano tre fondamentali tipologie di motrici illustrate in figura 2.1:



(a) Motrice serie 6000



(b) Motrice serie 5000



(c) Motrice serie 2800

Figura 2.1: Tipologie di motrici

La **serie 2800** è la rielaborazione dei tram più vecchi e diversificati tra loro. La *prima serie* fu costruita intorno al 1960 dalle **officine Stanga, Moncenisio e Seac** caratterizzata da una velocità di punta di 48 km/h, dotati di 3 porte, 27 posti a sedere ed un sistema di apertura e chiusura della porta semiautomatica dove con una leva il manovratore azionava la porta che a sua volta si apriva esclusivamente se il passeggero era sopra una pedana localizzata vicino alla porta e veniva segnalato al manovratore con una spia bianca in accensione. Questo sistema era molto pericoloso perchè non appena il passeggero scendeva dal secondo scalino la porta si richiudeva immediatamente alle sue spalle ed il manovratore era sprovvisto di specchietto retrovisore e non poteva verificare se avvenissero incidenti

durante la discesa. Nella postazione anteriore era posto il sedile del bigliettaio e la discesa avveniva nei pressi della porta posteriore, motivazione per il quale negli anni è rimasta la comune abitudine di salire da davanti e di scendere di dietro. La *seconda serie* fu invece costruita esclusivamente dalle **officine Seac** intorno al 1980, dove la nuova serie 2800 fu dotata di 4 porte e la velocità di punta fu ridotta a 45 km/h. La lunghezza del tram è approssimativamente di 20 m. Il tram è provvisto di sei assi che lavorano a coppie dove subito a valle del pantografo è posizionato un interruttore extrarapido in serie ad un interruttore di linea. Una rete di resistenze poste a monte rispetto ai motori consente tramite un inseritore manuale a tacche, che la corrente di avviamento non sia troppo elevata durante la fase transitoria. L'inversione della marcia è gestita tramite un sistema di contattori che escludono il motore su di un asse per alimentare quello opposto sull'altro asse [5].

La **serie 5000 T.P.R.** (Tram a Pianale Ribassato) fu introdotta circa nel 1990 e costruita dalle **officine Stanga**. Il vantaggio fondamentale di questa vettura è la mancanza di gradini per l'accesso ai disabili e alle carrozzine. La vettura raggiunge una lunghezza approssimativa di 22 m, con 51 posti a sedere, con una velocità di punta di 60 km/h ed una tensione di alimentazione di 600 V. L'equipaggiamento elettrico è caratterizzato da azionamenti a chopper con tiristori GTO raffreddati al freon. Il chopper controlla e regola la corrente di avviamento e di frenatura del motore di trazione in base agli ordini imposti dal manovratore attraverso il banco di manovra sottoposto a blocco meccanico e manipolatore a leva per effettuare manovre di marcia. Dal pantografo si ha una derivazione di 2 circuiti identici ed indipendenti tra loro ognuno dei quali va ad alimentare un gruppo motore di trazione composto da un motore serie a quattro poli. Ogni circuito si suddivide in 4 sottogruppi funzionali ovvero: una linea di alimentazione comprensiva di organi di protezione, una linea di filtro per attenuare le armoniche di corrente trasmesse in rete e ridurre il ripple di corrente nel motore, una colonna chopper e relativo motore di trazione ed un circuito di resistenze tampone e di frenatura. I contattori sono coordinati al fine di aprire in caso di sovracorrenti o sovratensioni. Lo scaricatore a valle del pantografo ha il compito di proteggere i circuiti contro le sovratensioni di origine atmosferica.

Dato che i raddrizzatori delle sottostazioni sono unidirezionali in corrente non è possibile reimmettere potenza nella rete di distribuzione MT ma può essere esclusivamente assorbita da altri tram che si trovano in una zona alimentata dalla medesima sottostazione elettrica. Motivazione per il quale la frenata rigenerativa è attuata in presenza di più tram con medesima alimentazione da sottostazione elettrica, viceversa si ha la frenatura di tipo reostatica con reostati posti sul tetto della vettura. In sintesi, il tram frena se la rete è recettiva ed immette potenza in rete, altrimenti la dissipa sulla resistenza. In caso di frenatura ho esclusione immediata del manipolatore a leva e in caso di guasti o anomalie nel funzionamento del chopper apertura automatica degli interruttori extrarapidi. La chiusura degli interruttori extrarapidi può essere svolta soltanto attraverso il manovratore dal banco di comando [5].

La serie **6000** detta anche *Cityway* fu costruita intorno al 2004 dalla **Alstom** negli impianti della Fiat Ferroviaria di Savigliano. La vettura dispone di 42 posti a sedere ed il pianale risulta totalmente ribassato non essendoci alcun asse che collega le ruote che risultano a loro volta totalmente indipendenti. L'alimentazione avviene in corrente continua a 600 V e la velocità di punta è pari a 70 km/h con avviamento non dissipativo attraverso l'utilizzo di convertitore statico di potenza dell'energia elettrica inverter. Subito a valle del pantografo è posizionato un contattore di linea, coordinato con un relè differenziale e con un fusibile, che permette la messa in sicurezza del sistema ed attraverso apposito regolatore di trazione e sezionatore il corretto sezionamento con messa a terra e la protezione da sovratensioni e cortocircuiti del convertitore statico dell'energia elettrica. Invece di utilizzare un interruttore extrarapido si sfrutta la configurazione fusibile più contattore perchè è più affidabile e si riducono i tempi di manutenzione. Per ottenere compatibilità di inserzione del convertitore statico dell'energia elettrica con la rete si interpone a monte del sistema un'induttanza di filtro in serie ed un sistema di condensatori e resistenze di scarica in parallelo, affinché la corrente assorbita di picco non superi i 1300 A riducendo la tensione ai capi dei condensatori al di sotto dei 50 V in corrente continua in tempi di scarica molto brevi. Si sfruttano anche un chopper per il sistema di frenatura e trasduttori per il controllo degli azionamenti [5].

Capitolo 3

Interfaccia al Simulatore di rete tranviaria software RHS di Genova

Scopo di questa sezione è fornire una panoramica del programma *Catasto rete di alimentazione* e delle applicazioni autocad realizzate dal *geometra Claudio Viale, Ivo Giammò* ed i tecnici del Servizio Lavori, che hanno fornito planimetrie e tutti i dati necessari per ricostruire la rete ed interfacciarsi al software della **RHS di Genova**. Il software sfrutta delle *autolisp* generate per la gestione, la manutenzione della rete e la collocazione geografica delle motrici. Le *autolisp* di Autocad prevedono la conversione diretta dei dati presenti in ambiente Autocad trasformati in ambiente Microsoft Access adatto per l'importazione nel programma **Catasto rete alimentazione** che a sua volta implementerà questi dati in un formato adatto all'elaborazione tramite software **Simulatore di rete tranviaria**. Si illustra il funzionamento del software tramite schema a blocchi di figura 3.1:

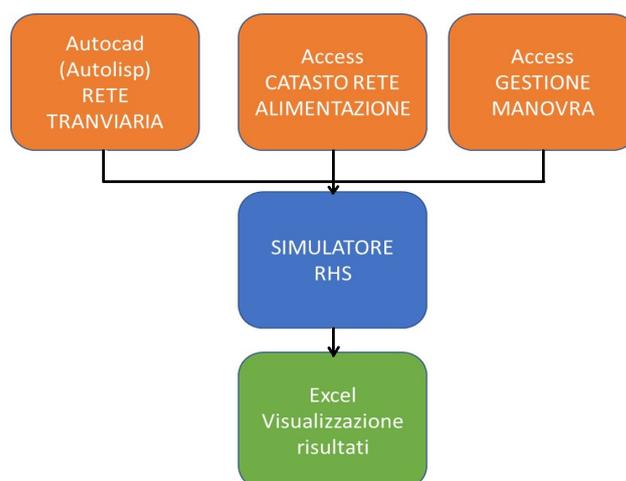


Figura 3.1: Schema concettuale funzionamento software Simulatore di rete tranviaria

3.1 Consultazione/editazione della rete

All'interno del software Autocad visibile in figura 3.2 è possibile inserire: motrici, cavi positivi, cavi di alimentazione, cavi negativi, cabine, pozzetti, cassette negative, lame per la connessione tra i cavi situati nei pozzetti e sui pali e nodi di linea aerea. Una volta inseriti tutti gli oggetti desiderati si procede all'esportazione dei dati in estensione *.out* apribile per visualizzazione tramite blocco note. Nell'esportazione la presenza della cartografia non è consigliabile perchè rallenta enormemente la velocità di esportazione. Si riscontra anche che le autolisp implementate con gli aggiornamenti successivi di Autocad inducono nel sistema delle anomalie nel funzionamento ed è necessario sfruttare un approccio differente per la risoluzione del problema che si pone.

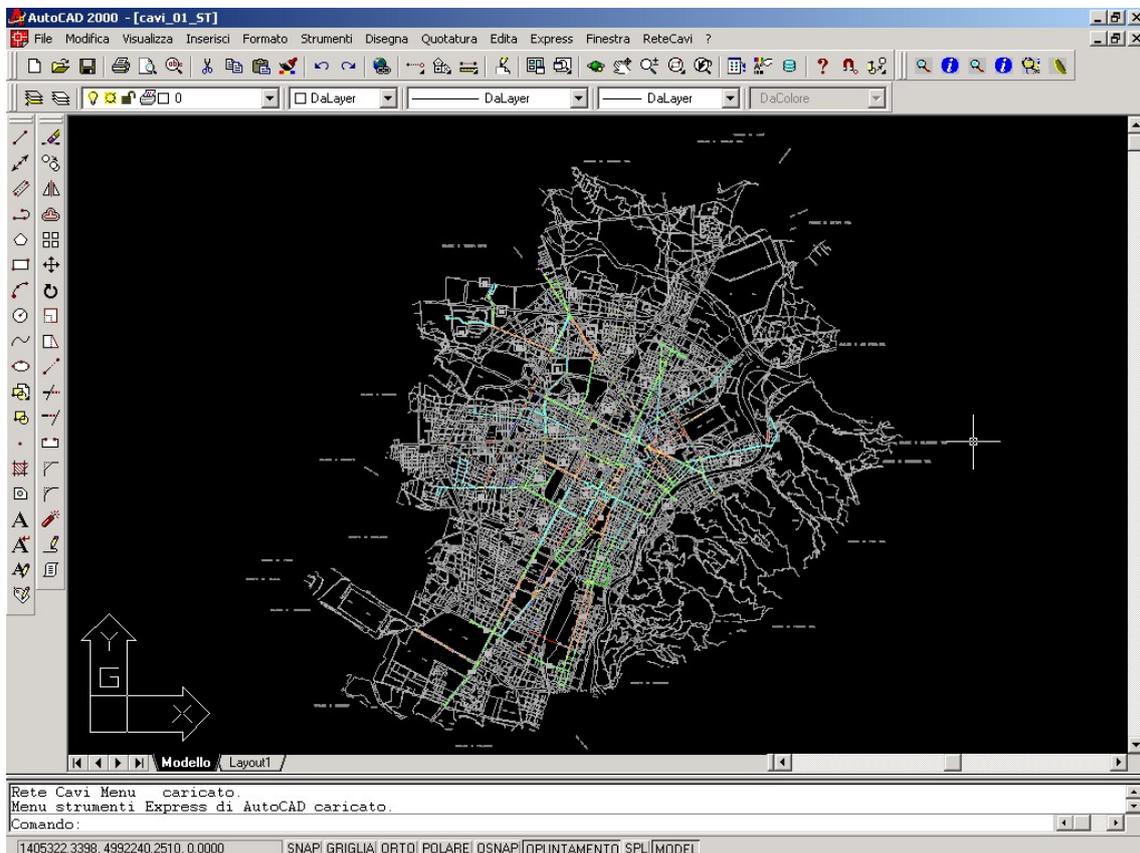


Figura 3.2: Schermata Software Applicazione Autocad

3.2 Importazione dei dati ed assegnazione della zona

La fase successiva è quella di importare i dati in estensione .out all'interno dell'ambiente Microsoft Access tramite applicativo Catasto rete di alimentazione visibile in figura 3.3. E' possibile possano effettuarsi errori dovuti ad eventuali difetti di sistema ed è preferibile effettuare necessariamente copia di back-up. Il software Catasto rete di alimentazione è realizzato in ambiente Microsoft Access tramite interfaccia programmata in Visual Basic e consente di ordinare dati, effettuare stampe con l'elenco dei cavi, delle cassette e di tutti gli elementi costituenti la rete di alimentazione, semplificare i circuiti, calcolare le resistenze dei cavi, unire due o più zone di alimentazione, importare o esportare file .dwg da Autocad ed esportare file per l'utilizzo sul Simulatore di rete tranviaria visibile in figura 3.4. Tramite la funzione importa da file Autocad si susseguono quattro fasi in cui nelle prime due si autorizza la procedura di aggiornamento dei record nel database, la terza fase accoda i dati e la quarta comunica che l'importazione è avvenuta correttamente. E' possibile applicare appositi filtri per importare determinate zone di alimentazione. Successivamente all'importazione dei dati nel database è necessario effettuare un'assegnazione manuale della zona ai cavi agendo all'interno del database tramite la funzione **Gestione manovra**. Tramite dei codici alfanumerici univoci riesco a definire connessioni, proprietà ed identificazioni di elementi connessi tra di loro. Una volta effettuate le assegnazioni manuali, chiuso e riaperto il database sarà possibile visualizzare l'aggiornamento delle modifiche apportate. Se tutte le connessioni sono state effettuate correttamente ed assegnate alla zona di alimentazione di appartenenza, è possibile effettuare l'analisi del circuito in una sezione del software dedicata. Terminato l'inserimento e l'elaborazione dei dati vengono le esportazioni in modo da consentire il calcolo del circuito da parte del simulatore di rete tranviaria in estensione .xls contenenti i dati dei nodi e dei lati della zona prescelta per il test. I programmi realizzati con Access ed Autocad tramite autolisp risultano efficaci, in sinergia con il simulatore di rete tranviaria, per la gestione della manutenzione, dello studio e della manipolazione della rete. In sintesi, grazie ai seguenti software di interfaccia è possibile: inserire vetture tranviarie distinguendone tipo e stato, interrogare dati anagrafici delle entità presenti in planimetria, ricercare varie entità tramite codici generati univocamente, esportare o importare dati, gestire la manutenzione della rete di alimentazione, gestire i vari layer di planimetria e gestire le stampe.

3.3 Calcolo e visualizzazione dei risultati

Il simulatore di rete tranviaria è costituito da un'interfaccia che sfrutta in background il programma MATLAB che costituisce il motore di calcolo di tutto il sistema. Mediante il pulsante di carica configurazione si effettua l'import dei dati precedentemente esportati e viene effettuato il calcolo della resistenza di ogni cavo. Successivamente attraverso il

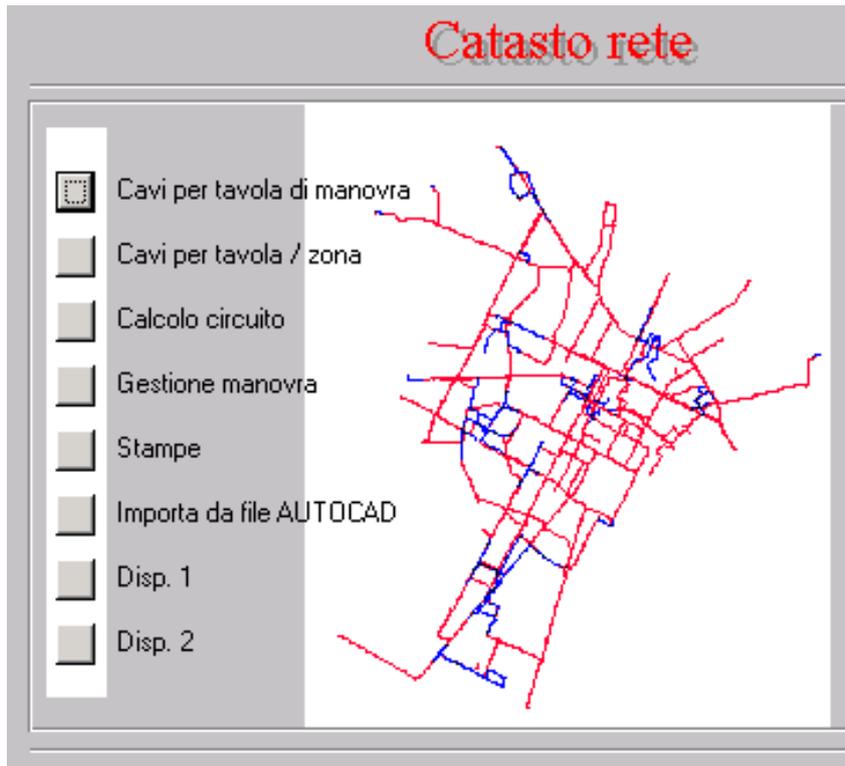


Figura 3.3: Schermata Software Catasto Rete di Alimentazione

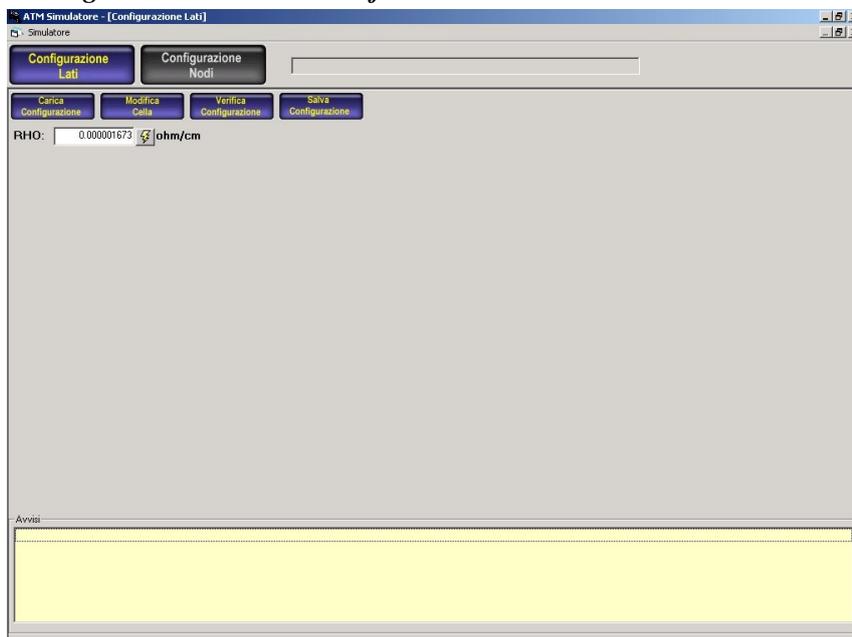


Figura 3.4: Schermata Software Simulatore di Rete Tranviaria

pulsante di configurazione nodi si caricano ulteriori dati relativi ai nodi e si può procedere alla simulazione avviando il processo relativo al calcolo delle correnti e delle tensioni della zona caricata. Al termine della procedura di calcolo si agisce sul pulsante risultati simulazione per la visualizzazione. Successivamente vengono esportati in ambiente Microsoft Excel i valori calcolati delle correnti circolanti nei rami e nei nodi e le cadute di tensione ai nodi. Qualsiasi modifica non può essere effettuata direttamente sul file esportato in excel, ma esclusivamente dal software di interfaccia Catasto reti di alimentazione, dal file con estensione textsl.out o dal simulatore di rete tranviaria. Rimane comunque il fatto che il committente desidera spostarsi verso *l'open-source* con un **software decisamente semplificato**. Viene rilasciato un file .dwg successivamente all'incontro nominato Agg_EL_cartografia.dwg creato in data 29 settembre 2009 contenente la planimetria di rete elettrica tranviaria della città di Torino in ambiente Autocad in riferimento alla figura 3.5.

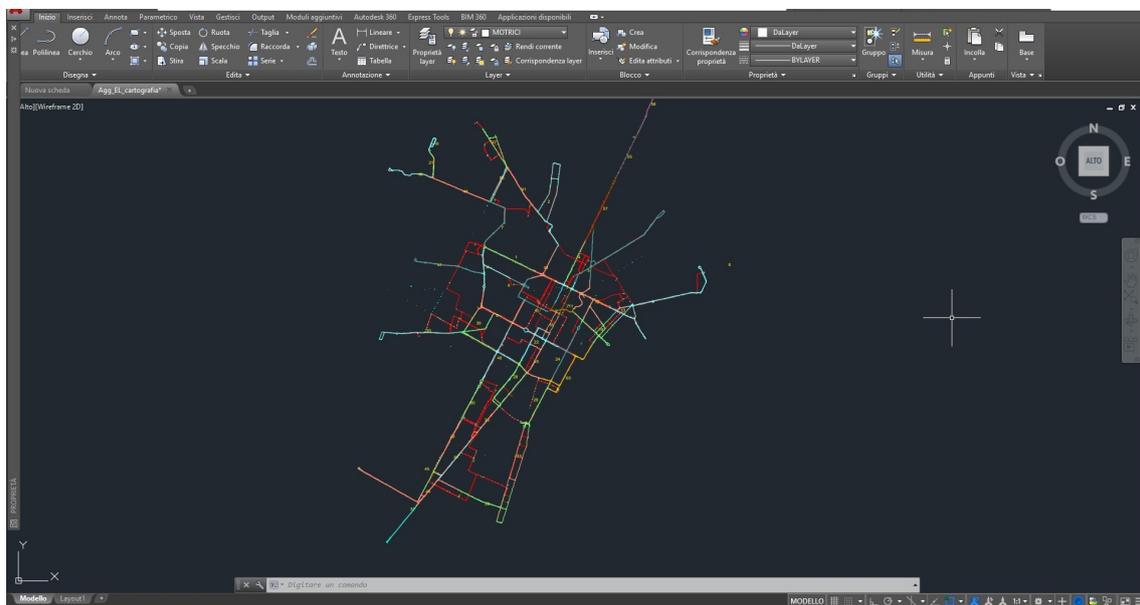


Figura 3.5: File *Agg_EL_cartografia Torino InfraTo*

Capitolo 4

Progetto 'Augusta Tauringrid'

Il committente richiede la possibilità di avere un ambiente di programmazione tale per cui si possano editare i blocchi, interrogare gli elementi individuando le proprietà di ciascuno di essi ed aggiornare facilmente qualsiasi tipo di modifica. Sulla base del file consegnato da **Infra.To** è stato possibile riscontrare preliminarmente che ciascuna tipologia di elemento risulta suddivisa in layer differenti e che le linee aeree risultano suddivise in zone e ciascuna di esse ha un suo layer specifico. Si decide di passare da ambiente Autocad ad ambiente QGIS e di effettuare esportazione in formato *.csv* e *.shp* tale per cui i dati vengono importati in Matlab e li elaborati per poi riportarli in QGIS ed effettuare visualizzazione grafica intuitiva e semplificata. Il codice sorgente verrà compilato in *.m* e si riporta uno schema a blocchi concettuale per la soluzione applicativa proposta in figura 4.1:

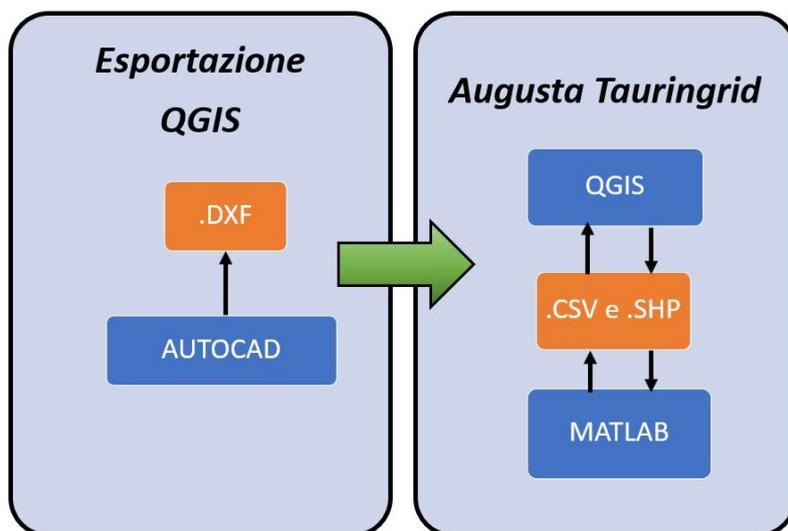


Figura 4.1: Schema concettuale funzionamento software Augusta Tauringrid

4.1 Esportazione da ambiente Autocad

In questo capitolo si illustrano sotto forma di manuale d'utilizzo tutti i vantaggi intrinseci del software e la loro reale applicazione per la parte sperimentale di calcolo. Si procede alla sperimentazione esportando vari tipi di dati appartenenti alla cartografia e ci si avvale dell'utilizzo della storia degli aggiornamenti e dei manuali di supporto per analizzare simbologie di forma e strutture utili all'identificazione di eventuali codici presenti. In seguito ad un incontro con i relatori si decide di esaminare l'ambiente di QGIS e analizzare se sia possibile l'importazione di dati Autocad senza particolari problematiche. Si procede all'esaminazione del software e si riscontra che è possibile importare vettori di dati al suo interno esportando i singoli layer da Autocad in formato *.dxf*. Si esegue la seguente procedura di analisi dell'ambiente di Autocad:

1. Si apre il file Agg_EL_cartografia in ambiente Autocad e si procede alla visualizzazione di tutti i layer presenti in maniera tale che non ve ne siano erroneamente alcuni nascosti che non possono essere esportati in ambiente QGIS come è possibile notare in figura 4.2.

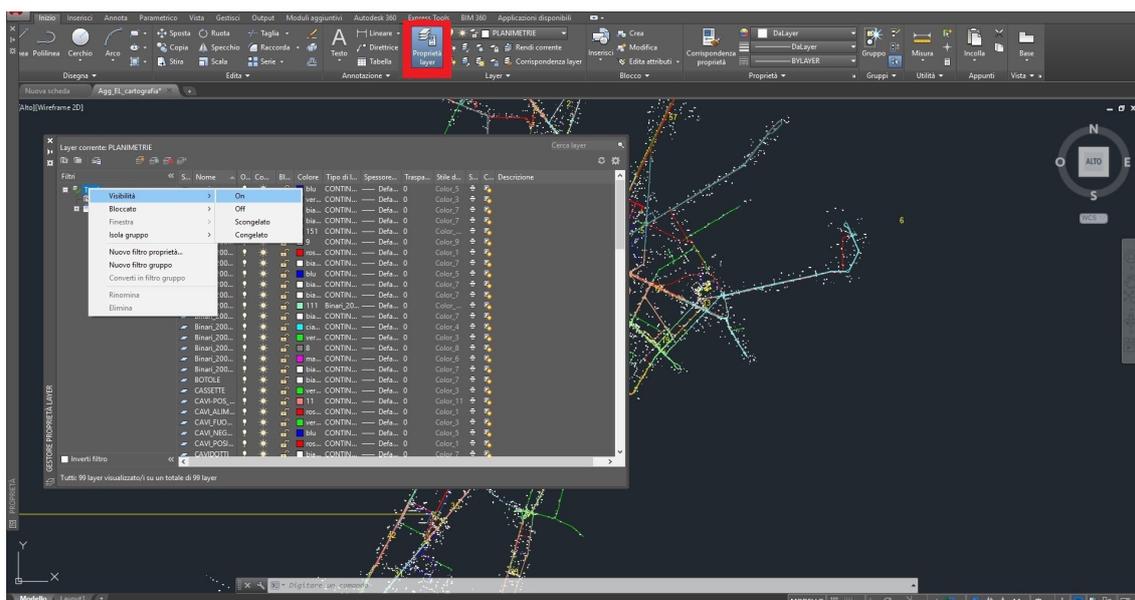


Figura 4.2: Step 1: Accensione visualizzazione di tutti i layer

2. Si procede con lo sblocco di tutti i layer presenti come illustrato nella figura 4.3.

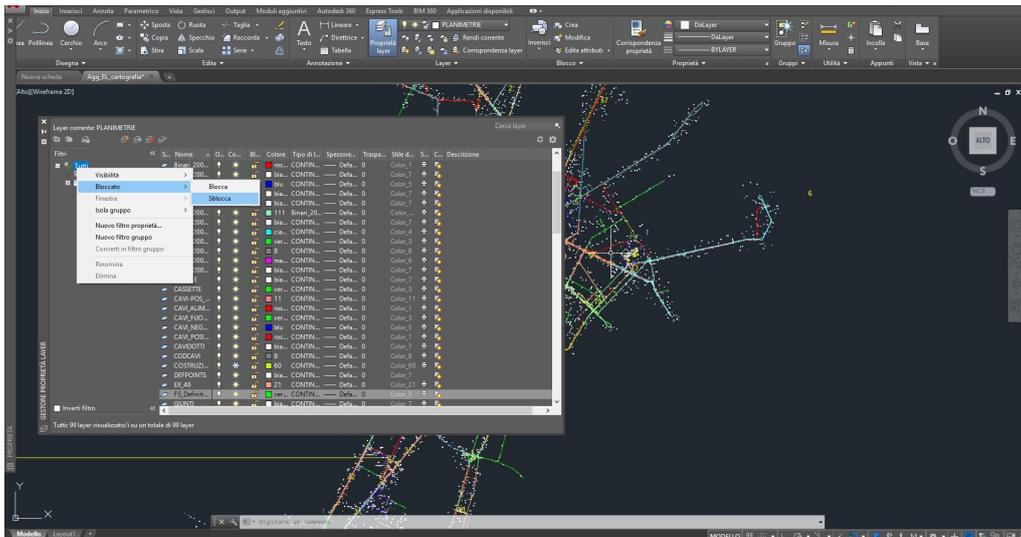


Figura 4.3: Step 2: Sblocco di tutti i layer bloccati

3. In seguito si scongelano i layer congelati notando che affianco al testo di scongelamento è possibile eseguire velocemente accensione della visualizzazione e sbloccaggio del layer tramite i tastini associabili a dei comandi rapidi visibili in figura 4.4.

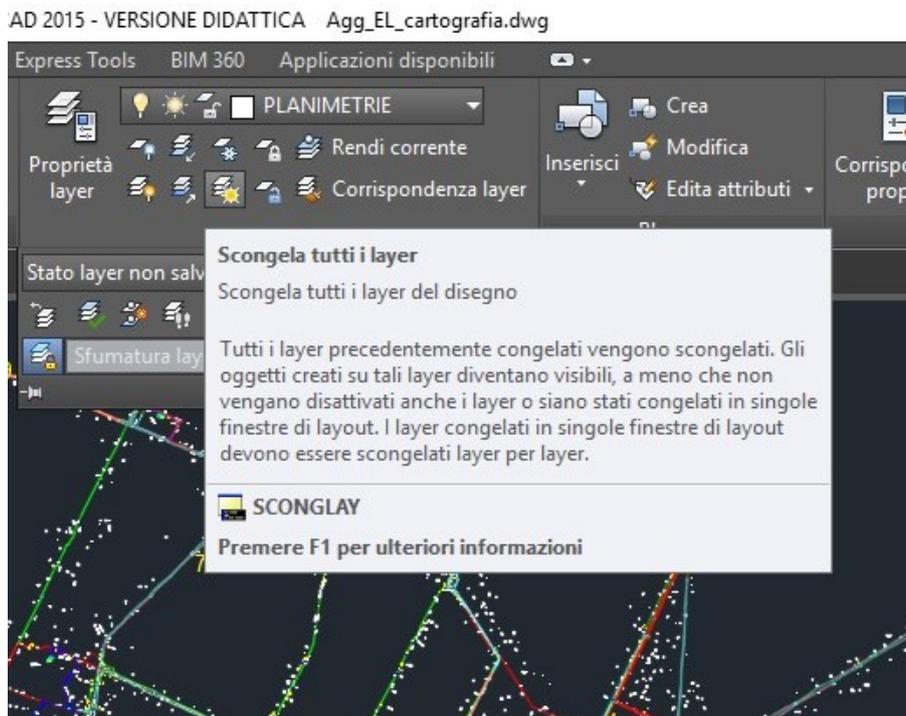


Figura 4.4: Step 3: Scongelamento di tutti i layer bloccati

4. Inoltre è possibile visualizzare dal file velocemente ciò che risulta di interesse tramite il comando Percorri i layer che tramite una procedura di filtraggio mi permette di verificare in maniera affidabile e veloce la presenza di determinati elementi presenti all'interno di specifici layer. Viene elencata la procedura corretta di esecuzione da figura 4.5 a figura 4.8:

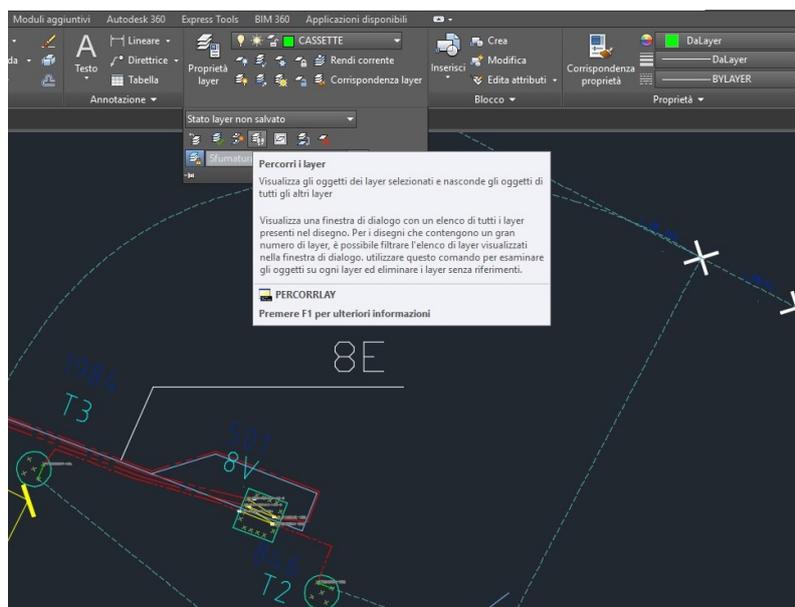


Figura 4.5: Step 4: Fase 1 - Comando Percorri i layer

5. Come fase iniziale bisogna duplicare il file ed aprirlo perchè bisognerà generare un .dwg per ogni layer. Bisogna procedere all'esportazione in formato .dxf di ogni singolo layer del file .dwg per l'importazione sottoforma di testo delimitato come vettore in ambiente QGIS e per farlo bisogna assicurarsi che siano attivi scongelati e non bloccati tutti i layer per poi bloccare il layer interessato e selezionando tutto l'insieme cancellare il resto affinché rimanga esclusivamente il layer interessato e salvarlo come *NOMELAYER.dwg*. Per puro esempio dimostrativo rimarranno non cancellati oggetti del layer **CASSETTE** sul nuovo disegno .dwg. Successivamente si procederà nell'illustrazione della corretta procedura di esportazione del layer in oggetto a titolo puramente dimostrativo a figura 4.9 e figura 4.10.
6. Per l'esportazione del layer selezionato in formato .dxf per l'importazione in ambiente QGIS, si decide di applicare la seguente convenzione come **standard univoco** per tutte le esportazioni che verranno effettuate salvando nel seguente modo: *NOMELAYER.dxf* per far ciò dal prompt dei comandi basta digitare *DXFOUT* oppure fare "Salva con nome" e specificare la tipologia .dxf della versione Autocad più aggiornata a seconda della compatibilità del sistema come mostrato in figura 4.11.

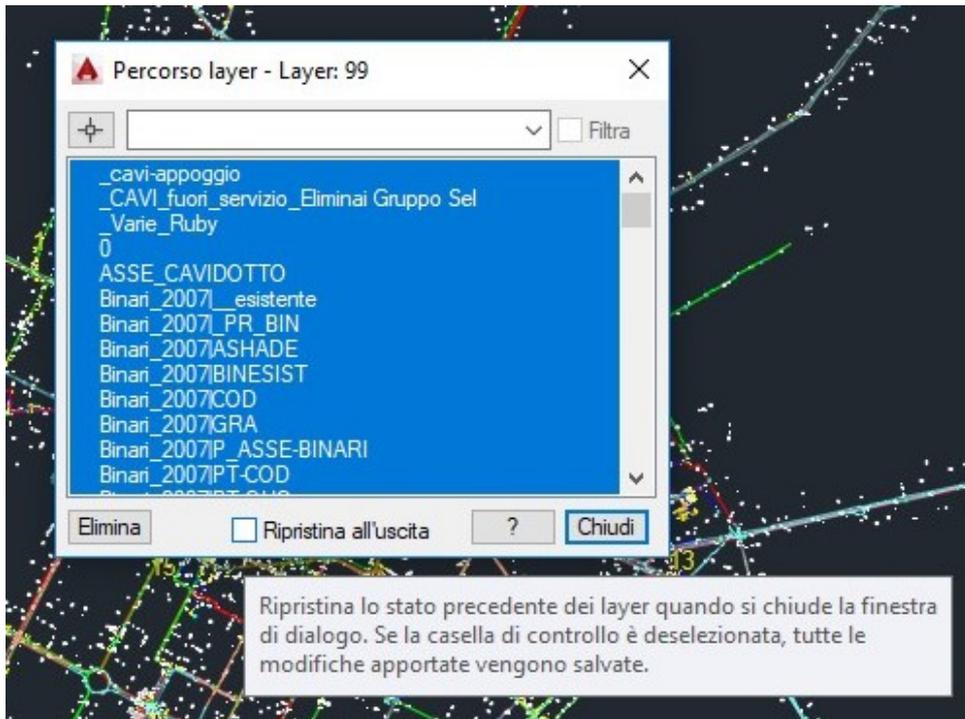


Figura 4.6: Step 4: Fase 2 - Comando Percorri i layer

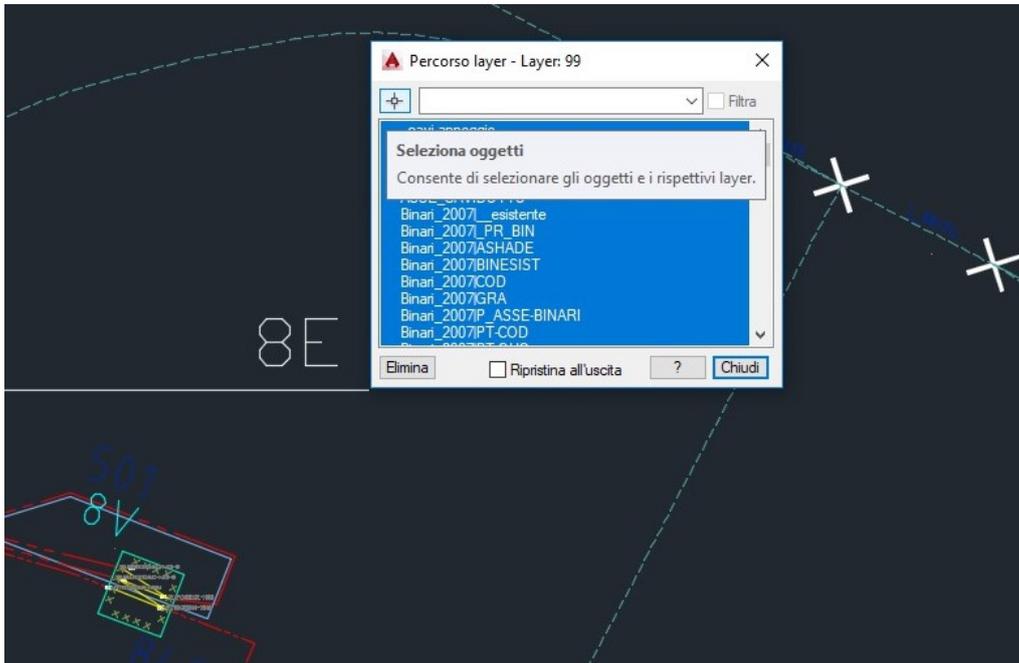


Figura 4.7: Step 4: Fase 3 - Comando Percorri i layer

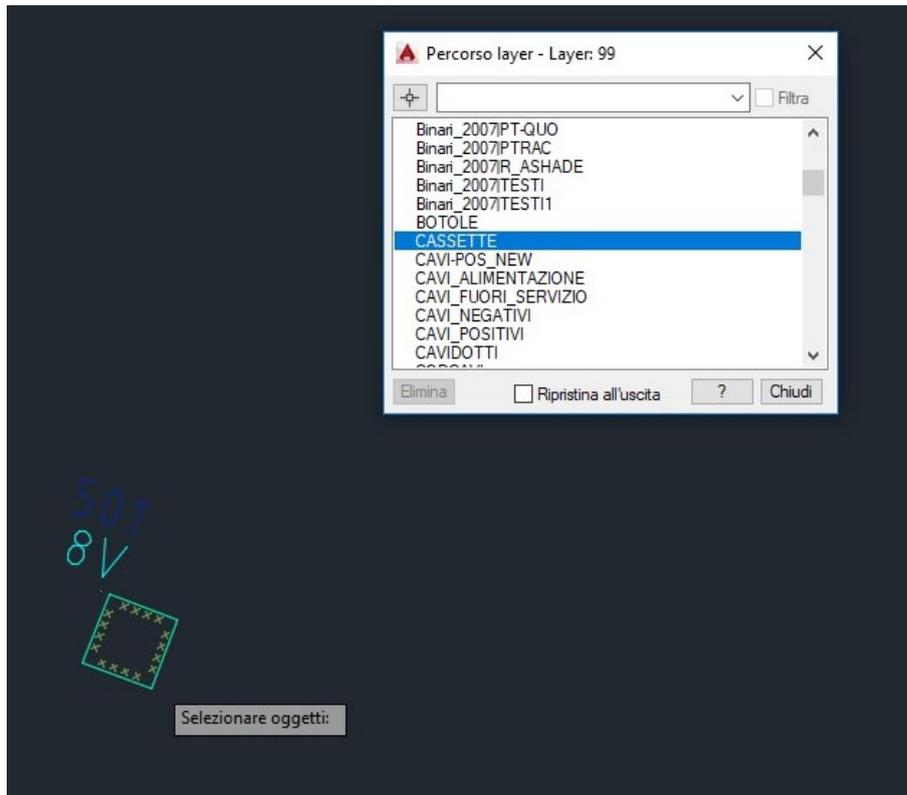


Figura 4.8: Step 4: Fase 4 - Comando Percorri i layer

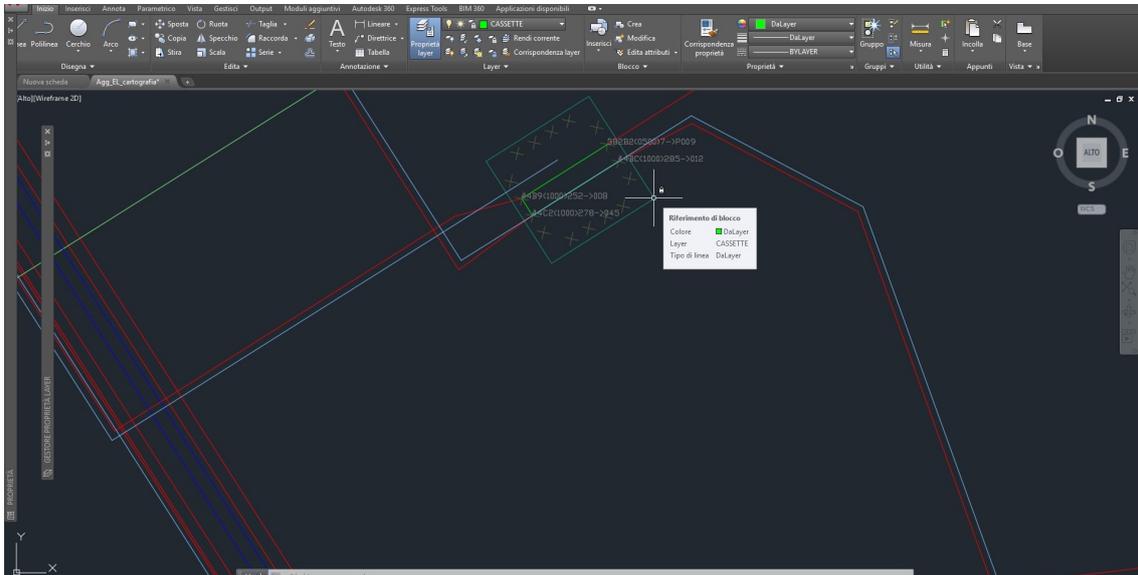


Figura 4.9: Step 4: Fase 5 - Comando Blocca

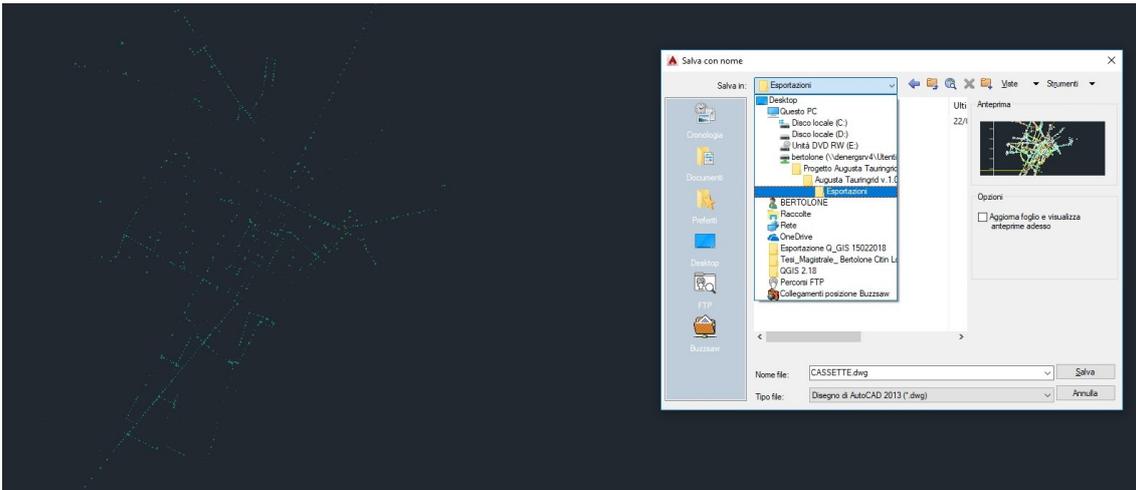


Figura 4.10: Step 4: Fase 6 - Salvataggio CASSETTE.dwg

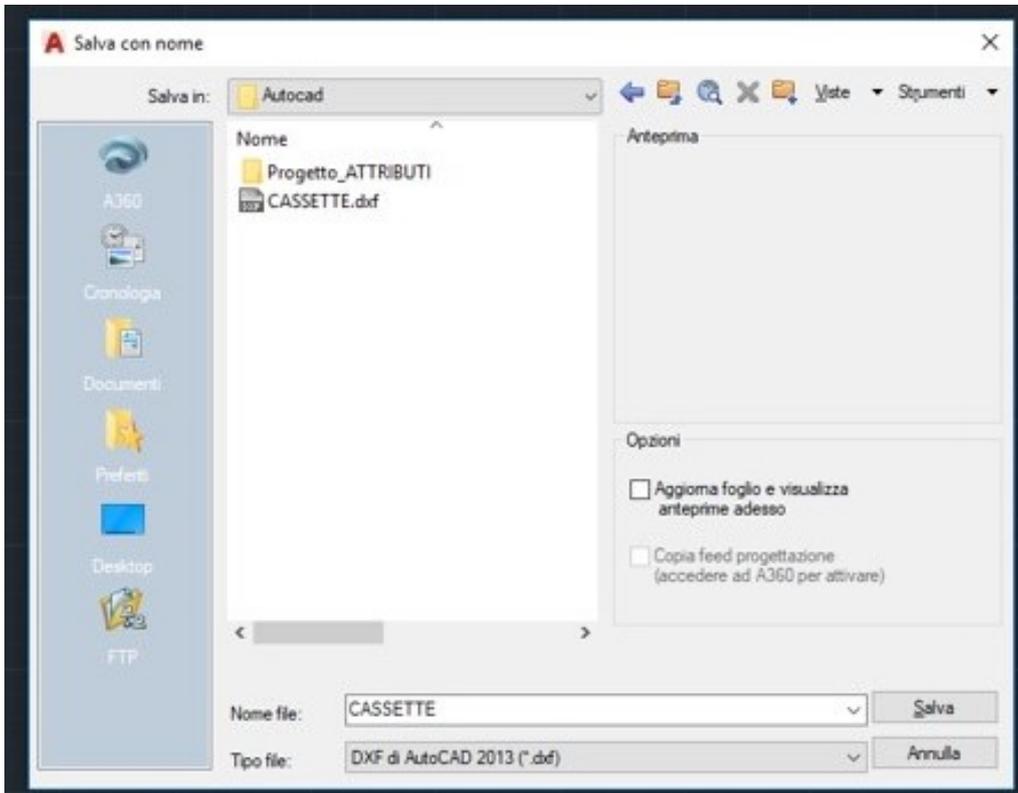


Figura 4.11: Step 5: Esportazione con estensione .dxf

7. Infine, una volta salvato il layer è sufficiente ripetere la procedura 5 e 6 per ogni singolo layer fino ad aver salvato ogni singolo layer in formato *.dxf*.

4.2 Configurazione plugin ed importazione in ambiente QGIS

L'ambiente QGIS risulta un ambiente ricco di plugin e versatile. Molte funzioni sono simili a quelle dell'ambiente Autocad dove è possibile generare linee, polilinee, circonferenze editare blocchi e molto ancora. Interessante è anche la funzionalità di snapping che mi consente di ottenere con estrema precisione connessioni tra elementi. L'ambiente di sviluppo si presenta con questa configurazione iniziale e necessita per completezza l'installazione di tre plugin scaricabili con la seguente procedura dettagliata. Sarà possibile per versioni del programma successivamente implementate aggiungere o modificare altri plugin a discrezione del committente. Si descrive la procedura dettagliata per l'installazione ed il settaggio iniziale dei plugin su calcolatore.

1. Effettuo attraverso la repository ufficiale l'aggiornamento dei plugin disponibili. Verifico che nelle impostazioni vi sia effettiva connessione online per scaricare i plugin direttamente dal sito ufficiale di QGIS. Nella figura 4.12 e figura 4.13, è possibile visualizzare l'interfaccia per la configurazione dei plugin.

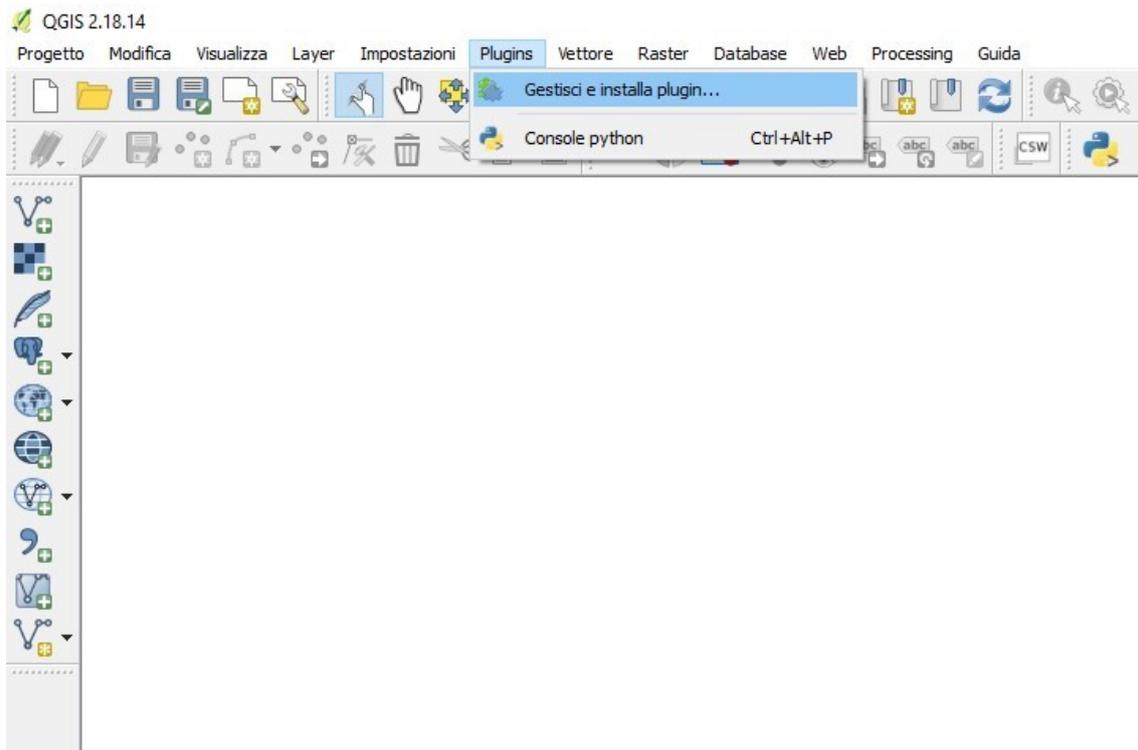


Figura 4.12: Step 1: Installazione Plugin

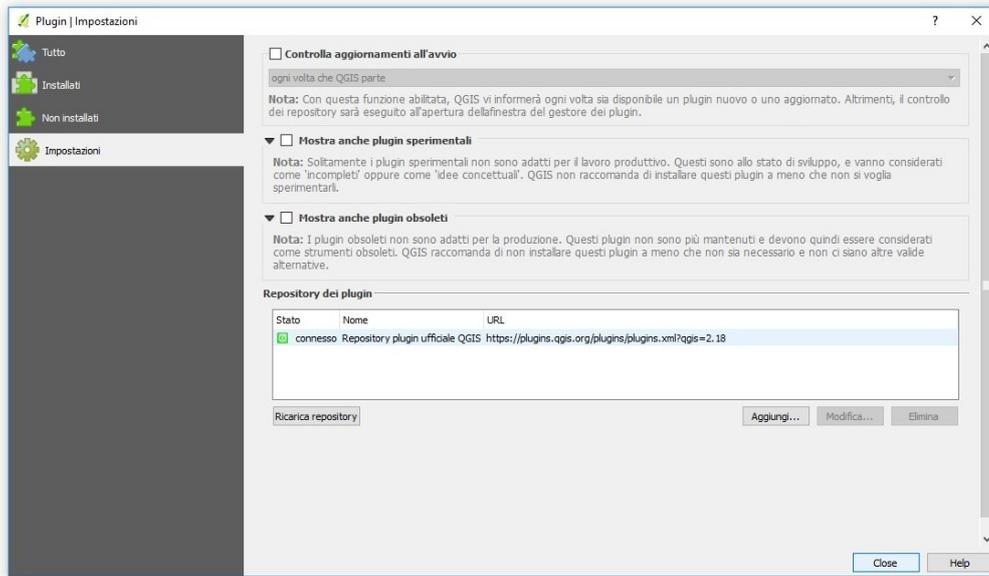


Figura 4.13: Step 2: Installazione Plugin

2. Installo tre plugin: **AutoFields**, **QuickExport** e **NNJoin** e noto che si aggiungerà una barra applicativa per ogni plugin installato sul pannello di interfaccia principale dell'applicazione visibile da figura 4.14 a figura 4.16.

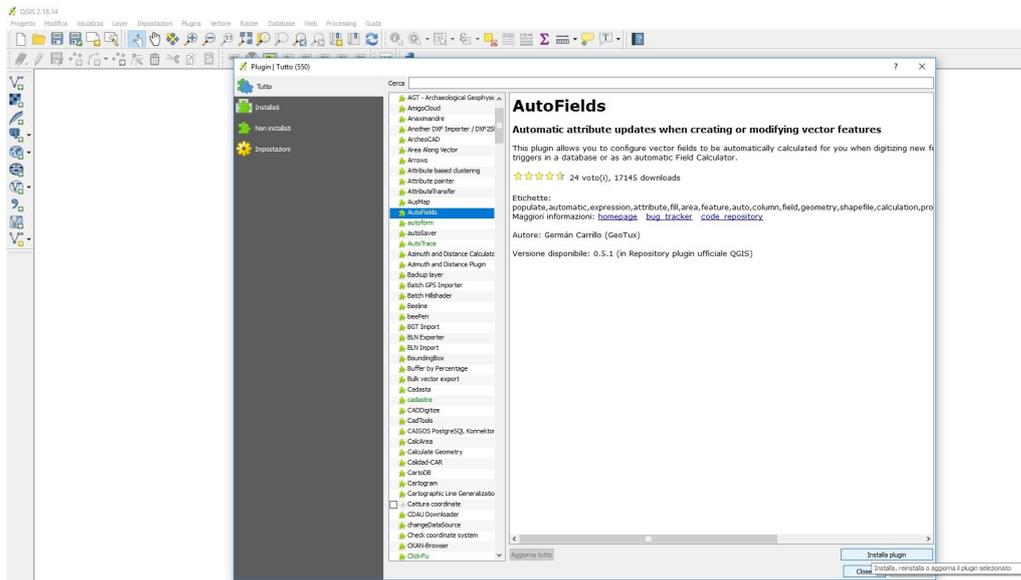


Figura 4.14: Step 3: Installazione Plugin

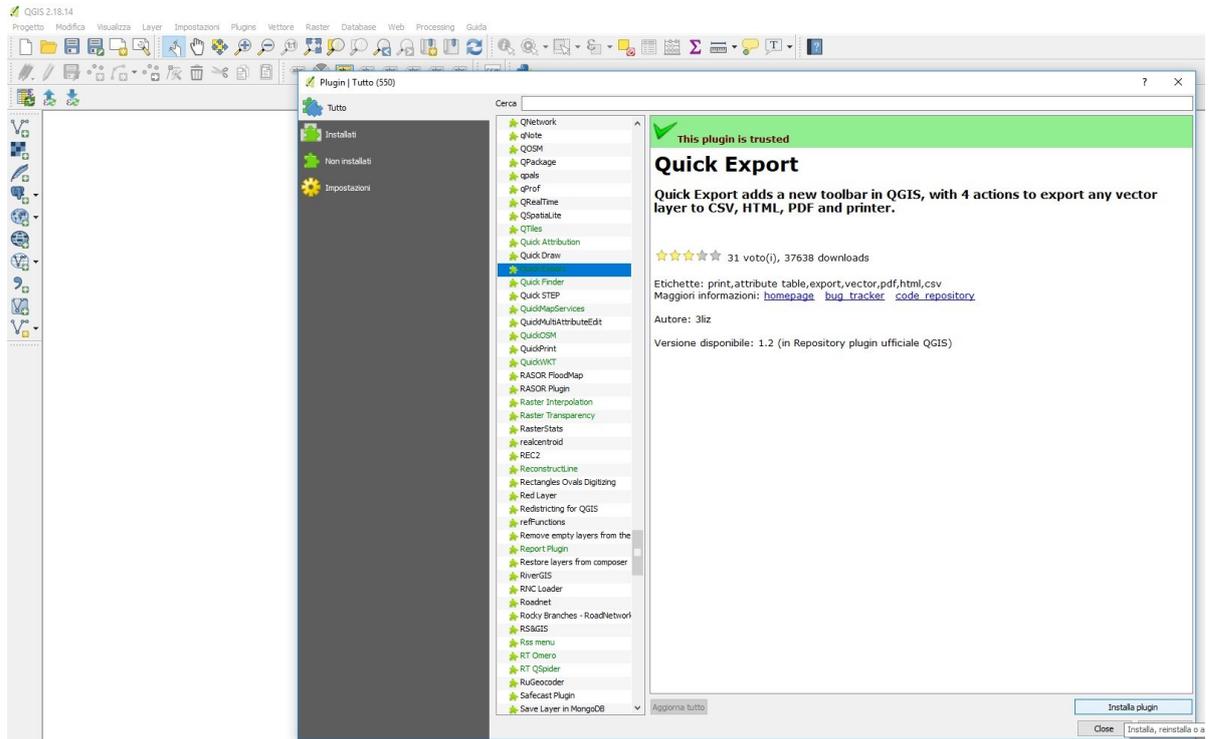


Figura 4.15: Step 4: Installazione Plugin



Figura 4.16: Step 5: Installazione Plugin

- Successivamente alla loro installazione mi occupo delle impostazioni di sistema dei plugin con una convenzione di lettura dei dati esportati univoca che si interfaccia in maniera ottimale con l'applicazione **Augusta Tauringrid** come visibile da figura 4.17 a figura 4.20.

4 – Progetto 'Augusta Tauringrid'

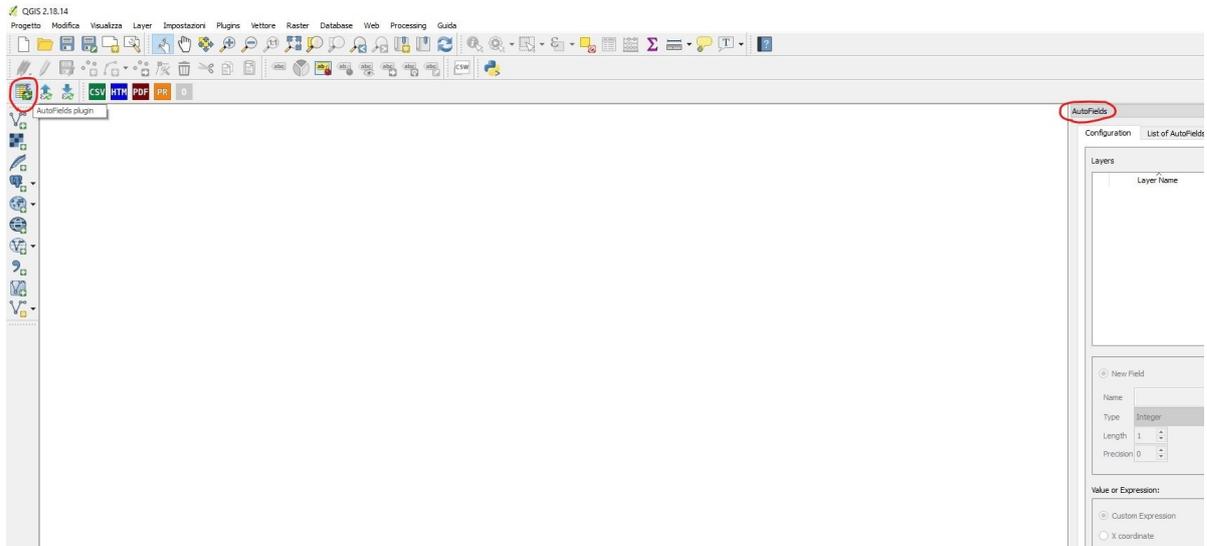


Figura 4.17: Step 1: Configurazione Plugin

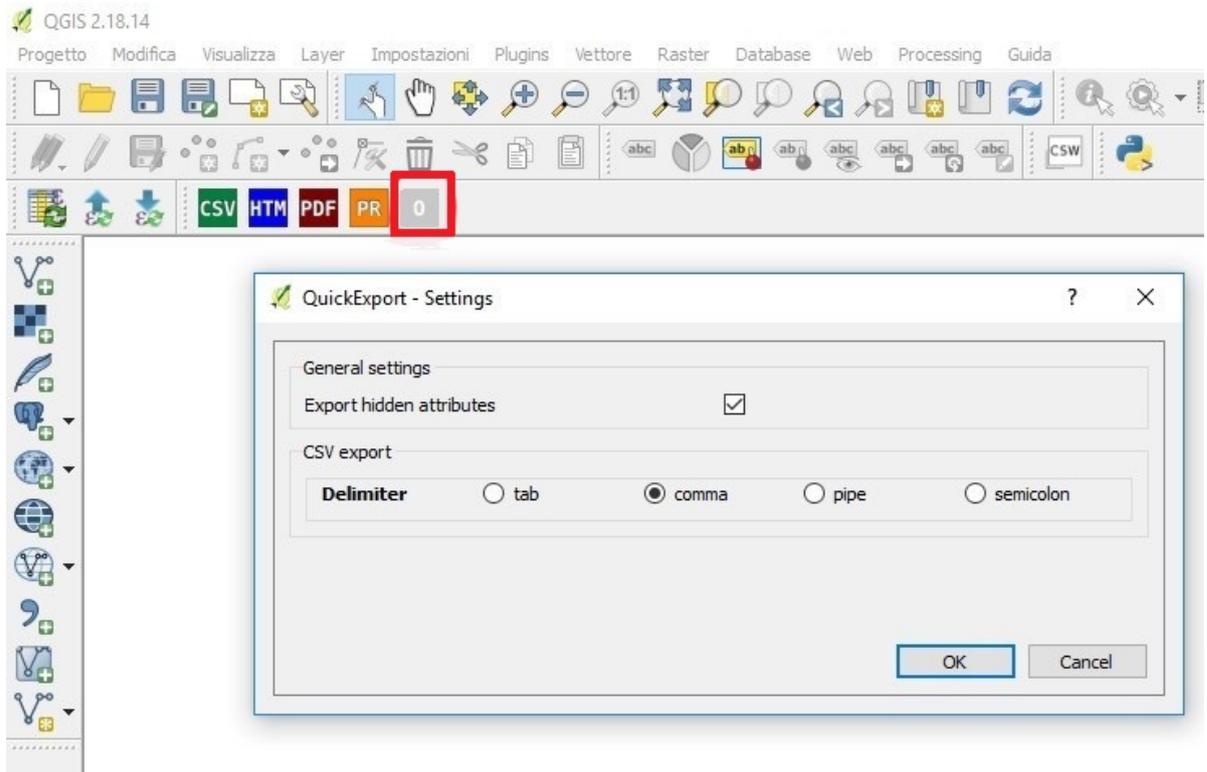


Figura 4.18: Step 2: Configurazione Plugin

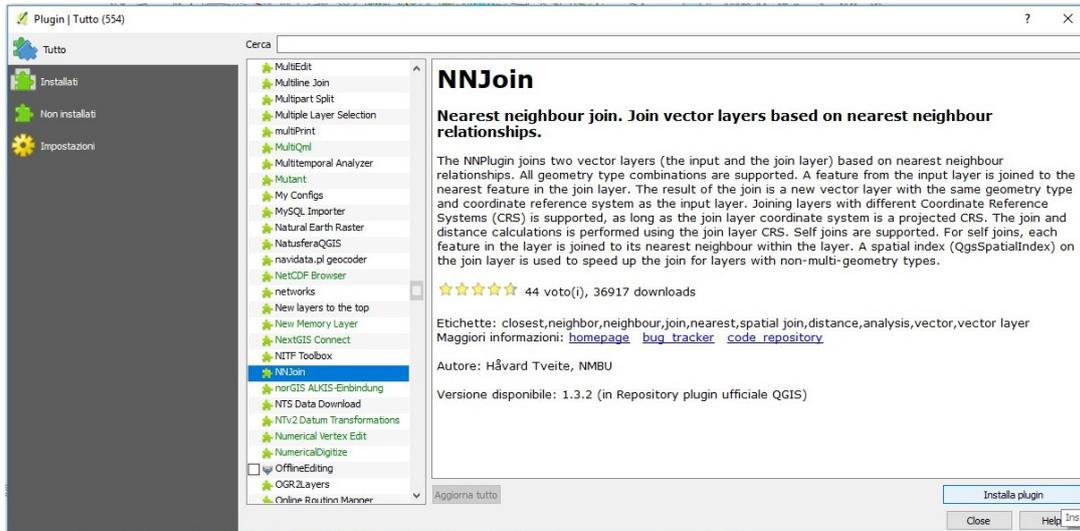


Figura 4.19: Step 1: Installazione Plugin

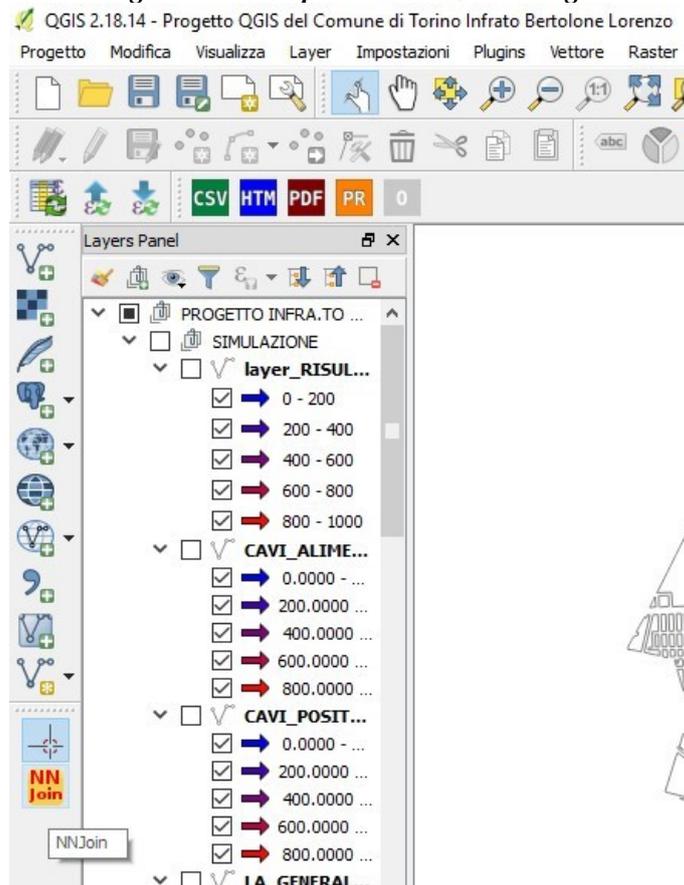


Figura 4.20: Step 2: Installazione Plugin

4. In seguito si provvederà a spiegare le funzionalità dei tre software installati per cercare di far comprendere al lettore la loro utilità. In prima analisi il plugin AutoFields è necessario per aggiornare automaticamente delle espressioni di campo formulate ed esportarle ed importarle in un formato *.json* standard. Il plugin QuickExport invece consente facilmente una volta selezionato un vettore di esportarlo in una cartella preimpostata con un determinato formato preimpostato ed estensione specifica che nel nostro caso sarà un *.csv* leggibile da Matlab. Infine il plugin NNJoin consente di associare in maniera corretta il tram alla linea aerea con l'ulteriore supporto della funzione snapping che verrà successivamente illustrata. E' necessaria l'esportazione del file *.csv* tramite il plugin NNJoin per associare il tram ad un punto di interesse della linea aerea durante la simulazione.

4.3 Elaborazione dati in ambiente QGIS

Una volta fatta una panoramica generale dell'ambiente QGIS si importano i file *.dxf* esportati precedentemente da Autocad tramite la seguente procedura e si illustrerà sempre esclusivamente a titolo esemplificativo la corretta procedura di impostazione per un unico layer. Gli altri layer da importare dovranno seguire la medesima procedura per effettuare una corretta importazione dei dati.

1. Bisogna preliminarmente inserire un nuovo layer in QGIS basandosi sul file precedentemente esportato e seguendo i sottopassaggi illustrati a figura 4.21 e figura 4.22.

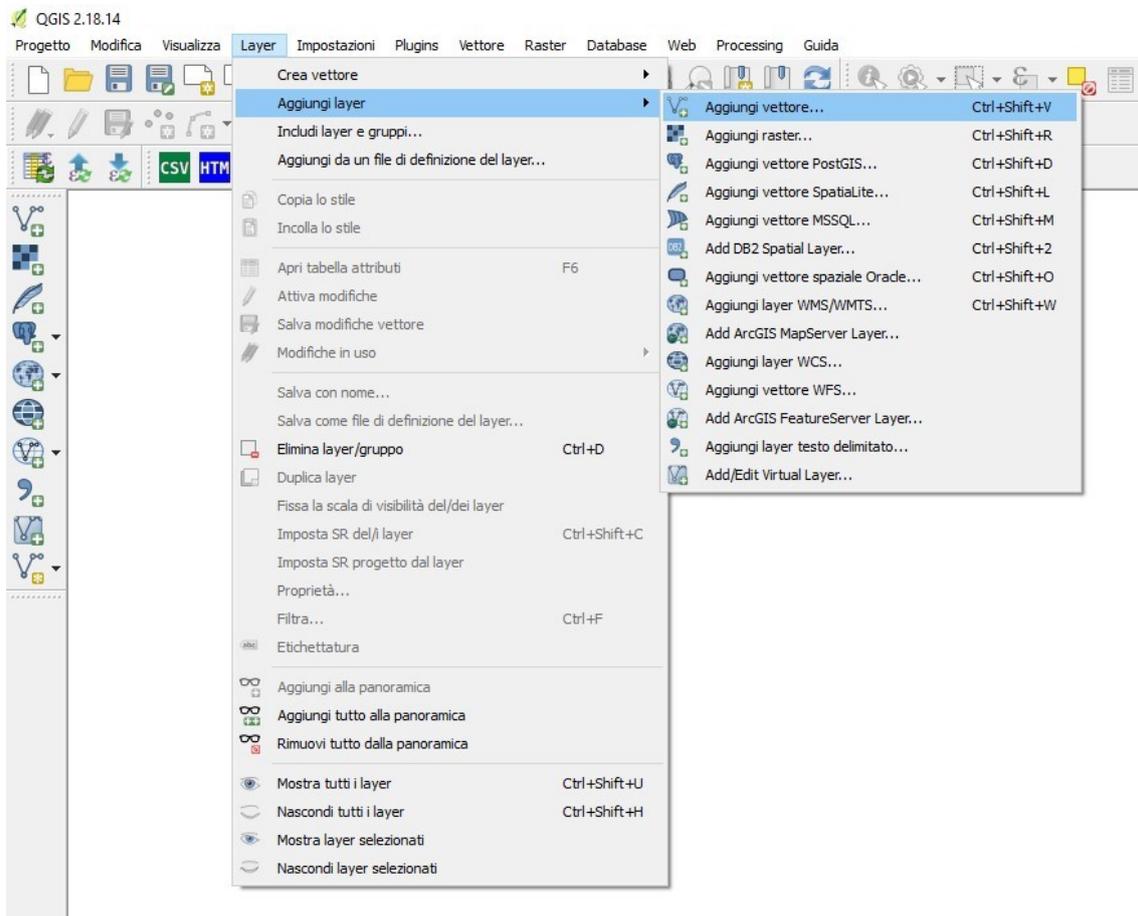


Figura 4.21: Step 1: Aggiunta di un vettore layer

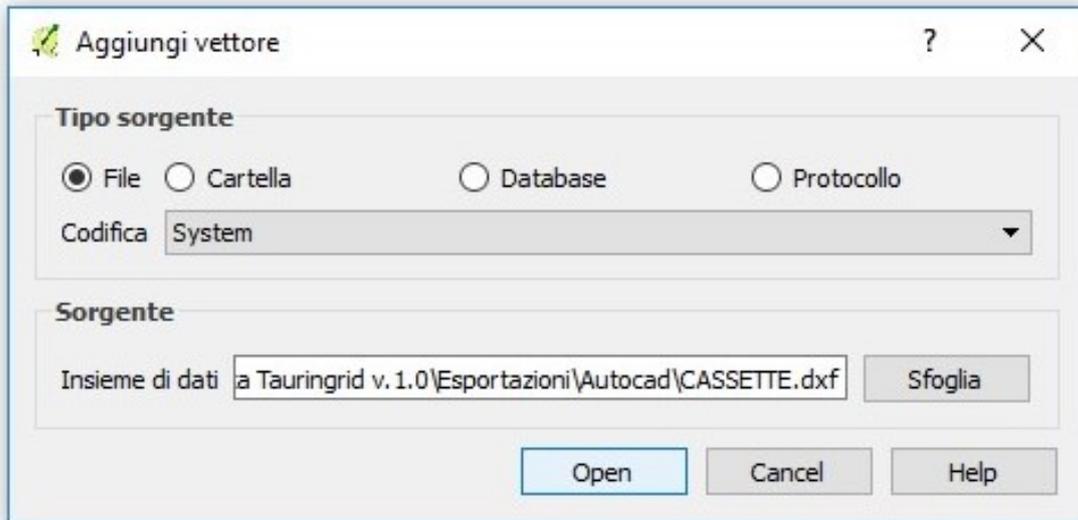


Figura 4.22: Step 2: Aggiunta di un vettore layer

2. Bisogna sempre evitare di inserire dati non necessari per alleggerire il programma. Qualunque layer importi introduco esclusivamente il tipo di geometria Point e LineString rispettando la convenzione di *sistema di riferimento* QGIS illustrata in figura sempre ed in qualunque caso venga richiesto come predefinito il **Monte Mario Italy zone 1** come è possibile notare in figura 4.23 e figura 4.24.

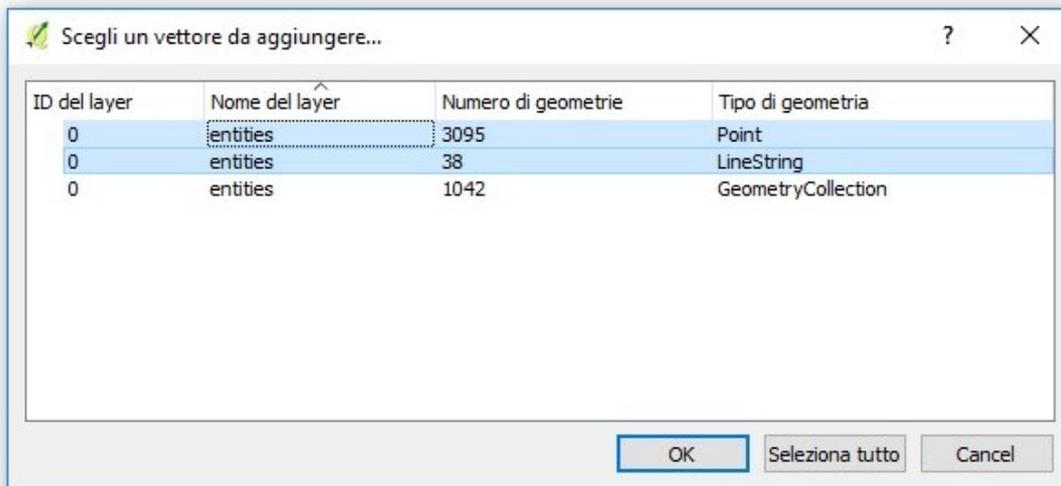


Figura 4.23: Step 3: Aggiunta di un vettore layer

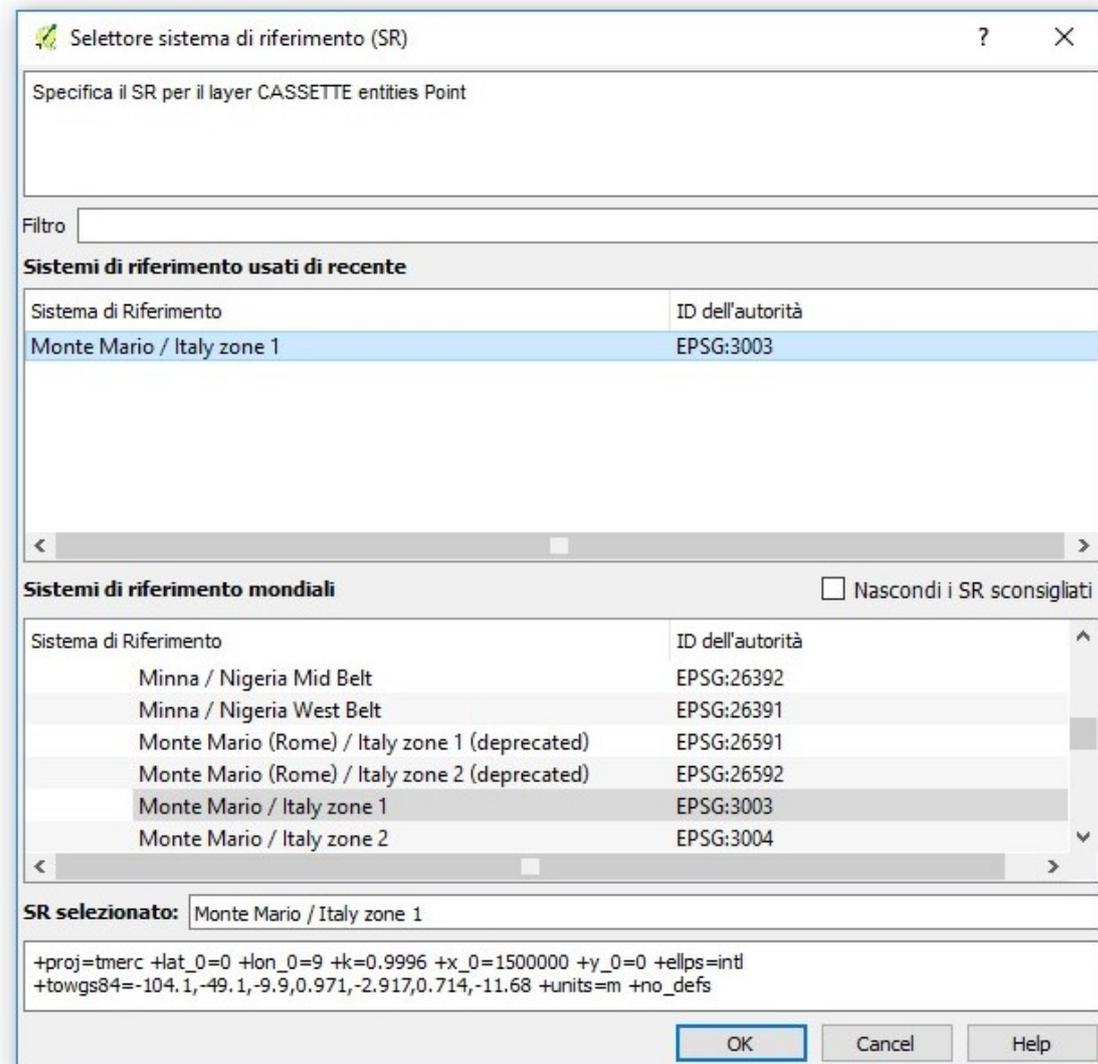


Figura 4.24: Step 4: Aggiunta di un vettore layer

- Una volta inserito il vettore nel pannello di visualizzazione dei layer è possibile visualizzare determinati elementi o meno e creare sottogruppi per dividere gli elementi in categorie di interesse. Per esempio è possibile realizzare una sezione simulazione che possa a sua volta includere simulazioni di zone facenti parte delle rete in un unico file effettuando confronti dato che ogni simulazione che verrà lanciata risulterà essere un layer sovrapponibile agli elementi già presenti nella rete. Inoltre è possibile selezionando il layer e con il tasto destro del mouse aprire la tabella degli attributi che potrà essere modificabile con aggiunta o rimozione di campi di calcolo in funzione delle esigenze del committente come visibile a figura 4.25 e figura 4.26.

4 – Progetto 'Augusta Tauringrid'

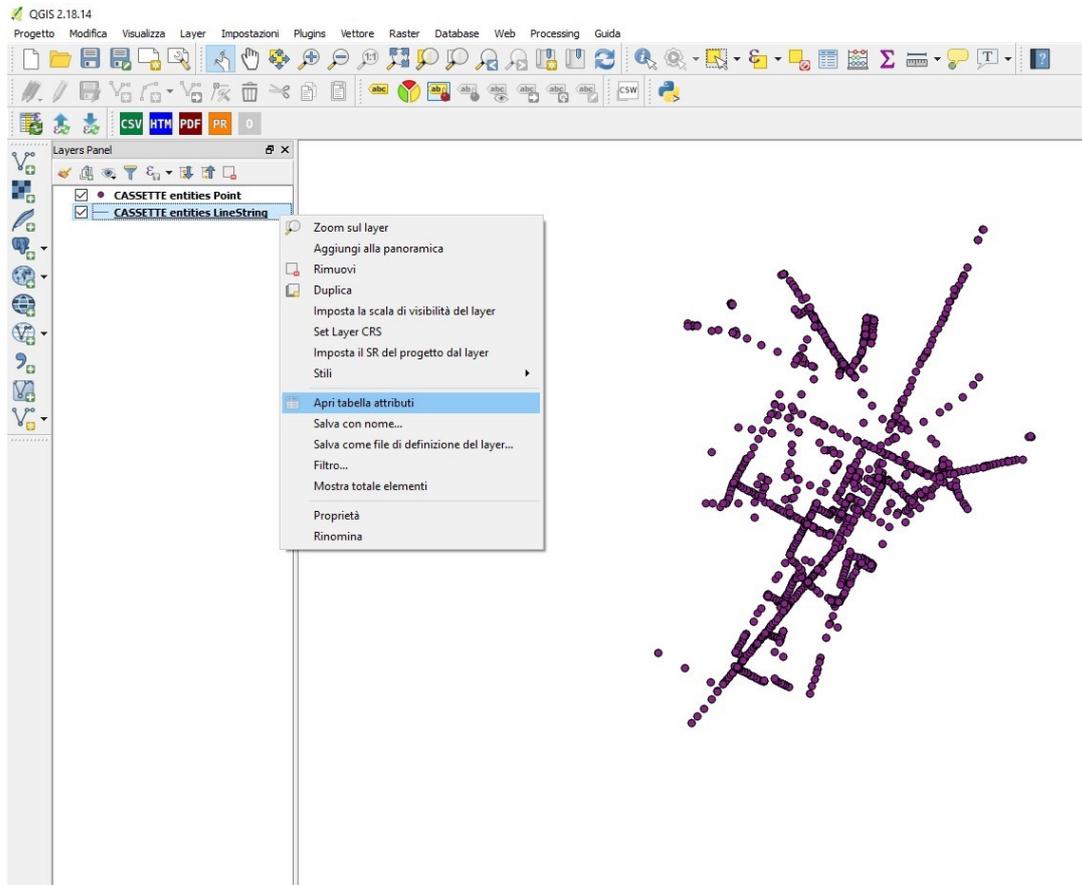


Figura 4.25: Step 1: Gestione di un vettore layer

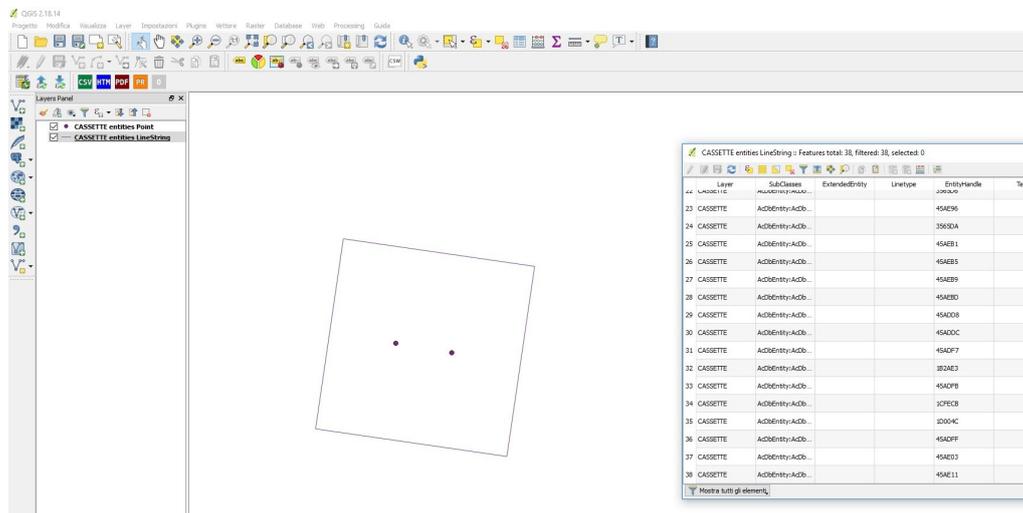


Figura 4.26: Step 2: Gestione di un vettore layer

4 – Progetto 'Augusta Tauringrid'

Il software come versione iniziale è stato impostato includendo al suo interno le cabine di alimentazione con nominativo e codice univoco, i cavi positivi, i cavi di alimentazione, le lame e le linee aeree. Al committente viene rilasciata versione QGIS già impostata secondo la procedura precedentemente illustrata e con successivi affinamenti aggiungendo campi di calcolo alle tabelle ed escludendo dati non necessari all'interno della cartella di sottogruppo extra. Si mostrano gli strumenti attraverso il quale è stato possibile svolgere ulteriori affinamenti sul file da figura 4.27 a figura 4.29:

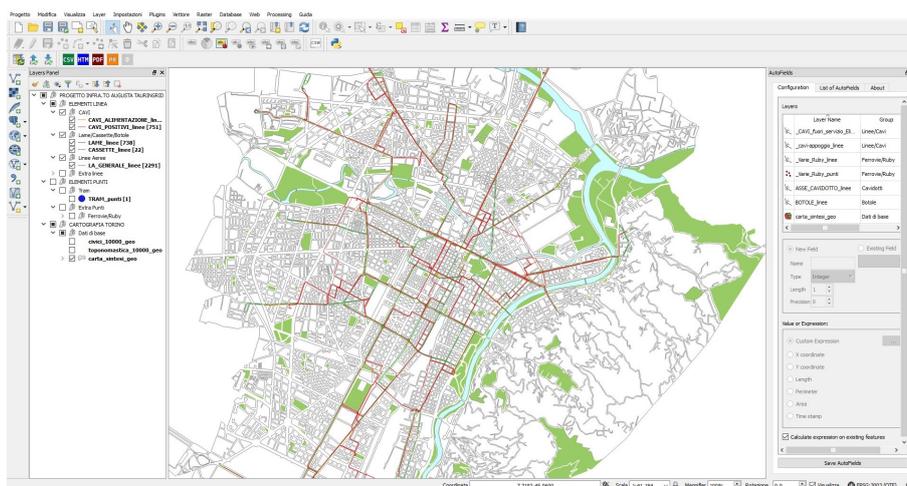


Figura 4.27: Panoramica Augusta Tauringrid 1

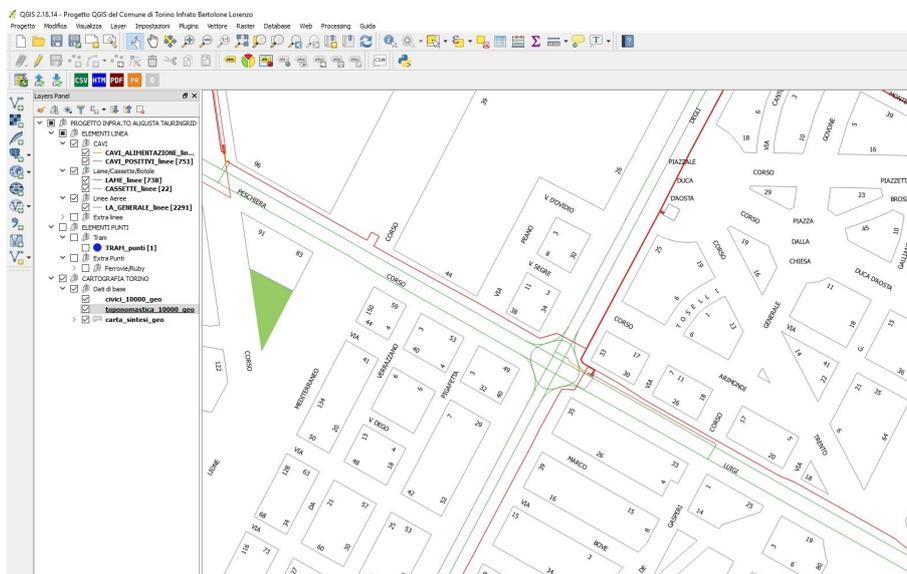


Figura 4.28: Panoramica Augusta Tauringrid 2

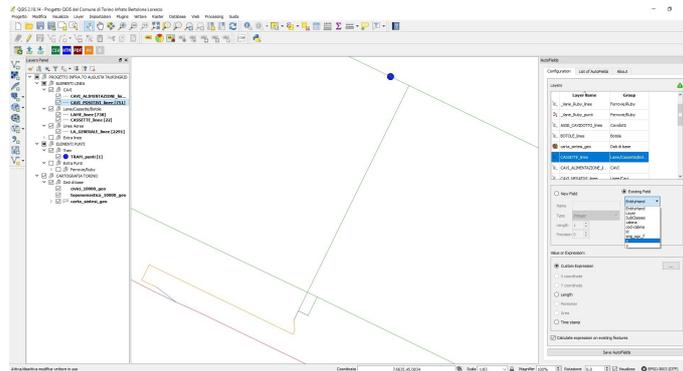


Figura 4.29: Panoramica Augusta Tauringrid 3

1. I campi vengono inseriti manualmente su AutoField sfruttando la seguente procedura guidata da figura 4.30 a figura 4.33 in funzione delle estrapolazioni di entità geometriche da elemento ad elemento. Una volta inseriti tutti i campi necessari si può procedere all'esportazione in un formato *.json* che sarà importabile esclusivamente la prima volta direttamente ove necessita il suo utilizzo.

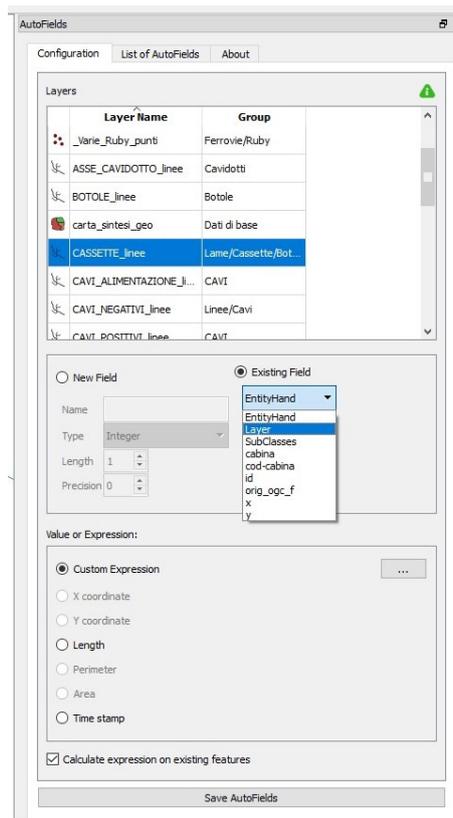


Figura 4.30: Step: 1 - Plugin AutoFields

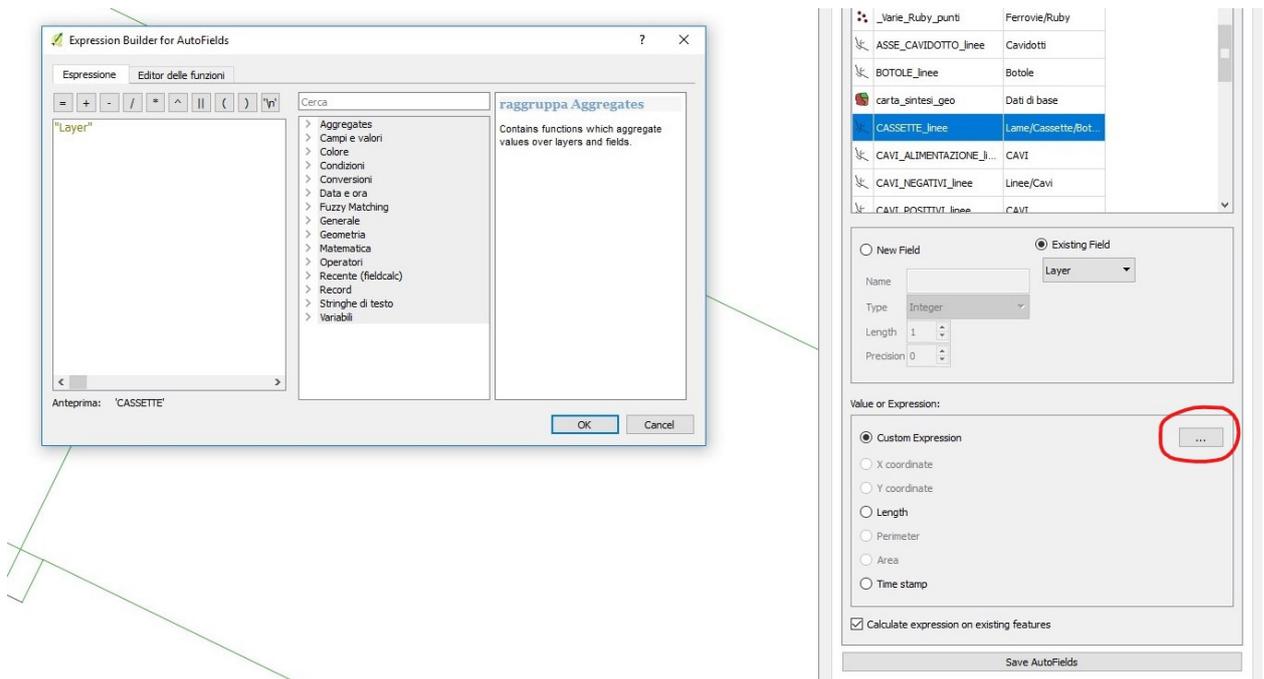


Figura 4.31: Step: 2 - Plugin AutoFields

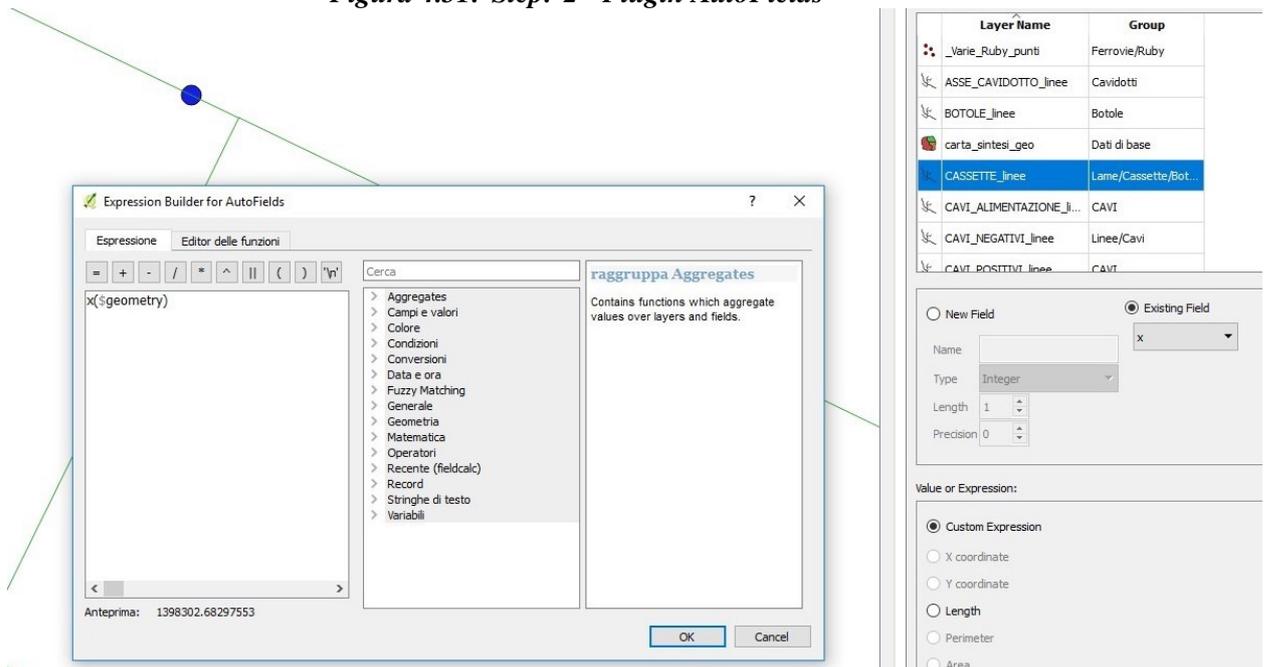


Figura 4.32: Step: 3 - Plugin AutoFields

LUNGHEZZA

```
length( $geometry )
```

INIZIO POLILINEA/LINEA

```
to_string(geom_to_wkt(start_point($geometry)) )
```

FINE POLILINEA/LINEA

```
to_string(geom_to_wkt(end_point($geometry)) )
```

SEQUENZA VERTICI LAME/BOTOLE

```
to_string( geom_to_wkt( nodes_to_points( $geometry )))
```

Figura 4.33: Step: 4 - Codici Plugin AutoFields

- Tramite il plugin NNJoin è possibile associare ciascun tram alla linea aerea di contatto più vicina e verrà creato un layer automatico di associazione che dovrà essere correttamente esportato durante la fase di esportazione dei dati .csv da QGIS a Matlab per le corrette elaborazioni. E' strettamente necessario nominare il file di output in maniera corretta come **layer_TRAM**. Il plugin in caso di modifiche strutturali alla rete o modifiche del layer TRAM_punti va rieseguito ed esportato nuovamente per l'associazione corretta tra i layer come visibile in figura 4.34 e figura 4.35.

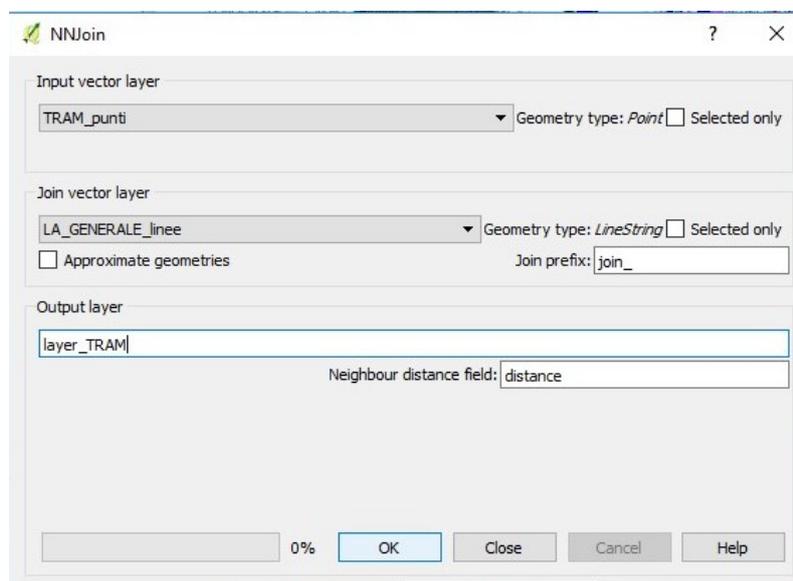


Figura 4.34: Step: 4 - Codici Plugin AutoFields

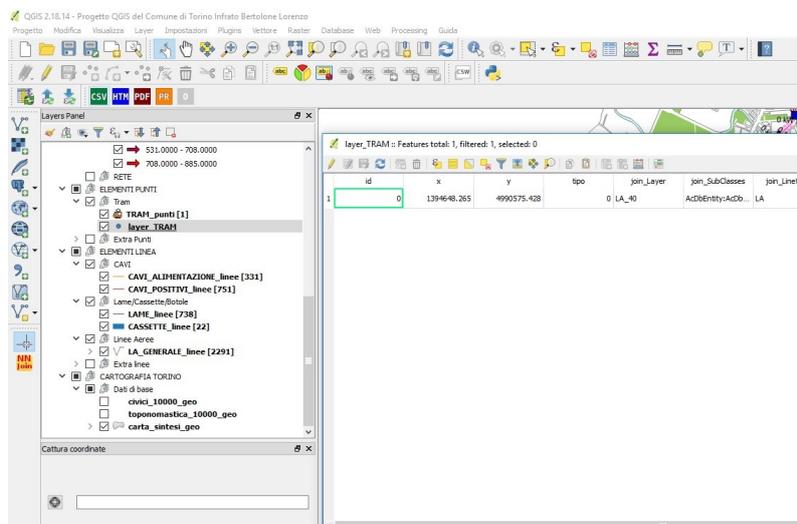


Figura 4.35: Step: 4 - Codici Plugin AutoFields

- Il software risulta molto utile per la modifica tabulare dei dati ma anche grafica tramite la possibilità di aggiungere entità, connettere elementi tramite snapping, effettuare misure o quotature in maniera diretta, modificare entità che si aggiorneranno automaticamente in tabella grazie al plugin AutoFields e per completezza vengono illustrati da figura 4.36 a figura 4.51. Viene effettuata una breve panoramica delle potenzialità del software oltre alla possibilità di ulteriore supporto grazie ad una vasta repository di documentazione aggiornata online.

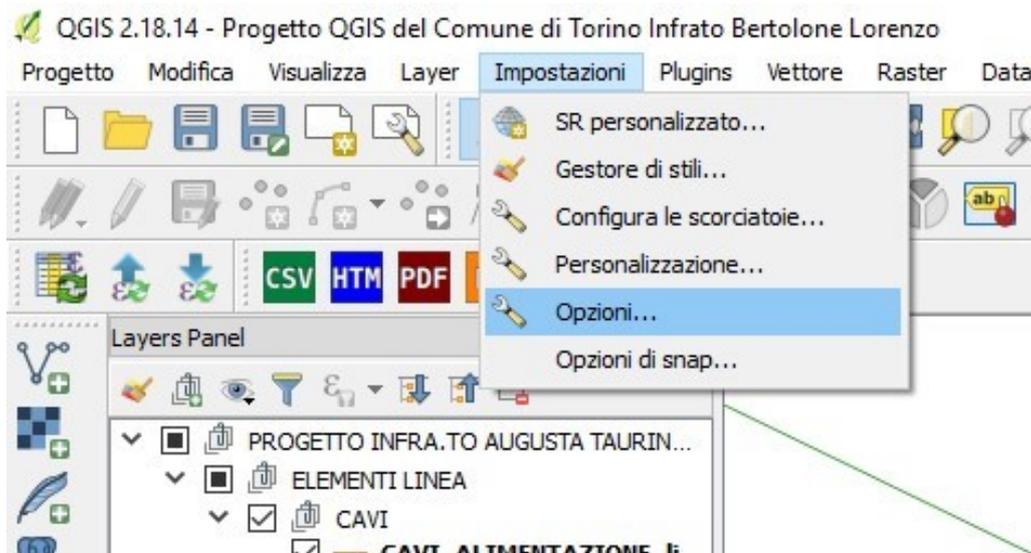


Figura 4.36: Step: 1 - Aggiunta elementi

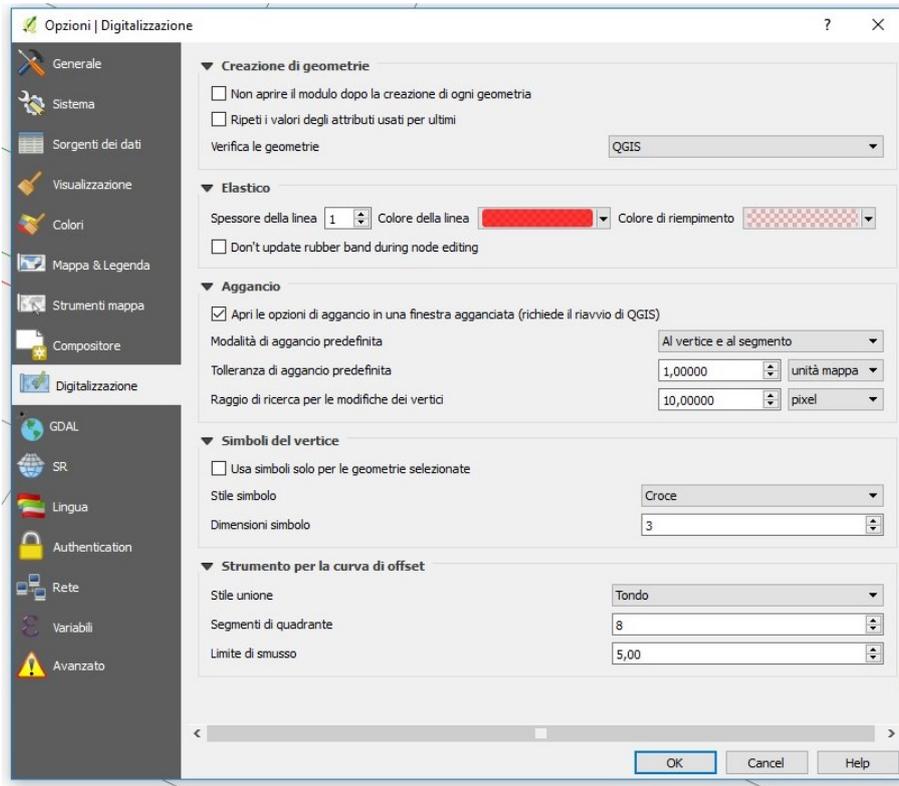


Figura 4.37: Step: 2 - Aggiunta elementi

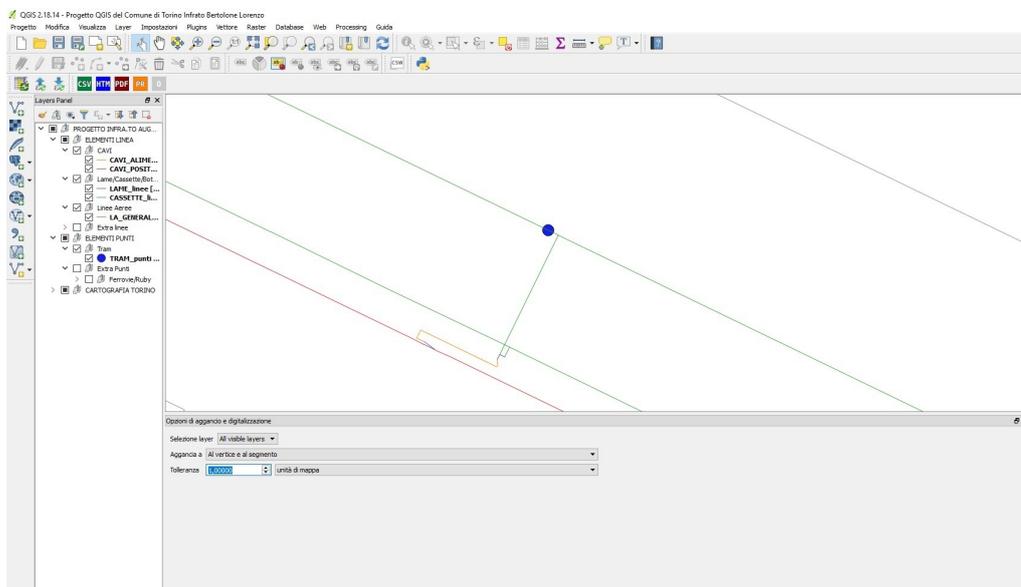


Figura 4.38: Step: 3 - Aggiunta elementi

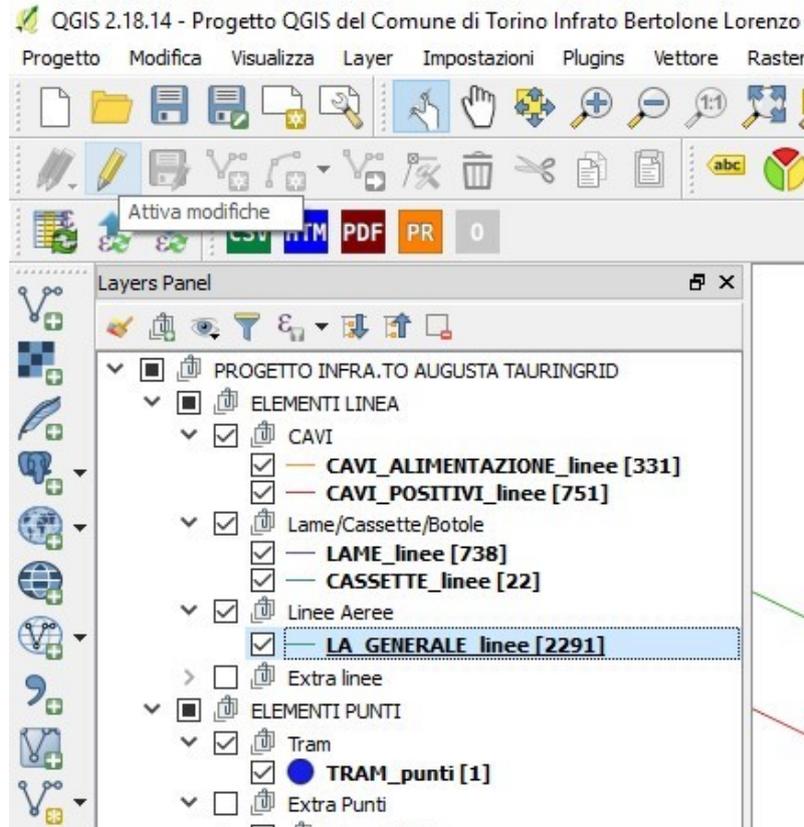


Figura 4.39: Step: 4 - Aggiunta elementi



Figura 4.40: Step: 5 - Aggiunta elementi

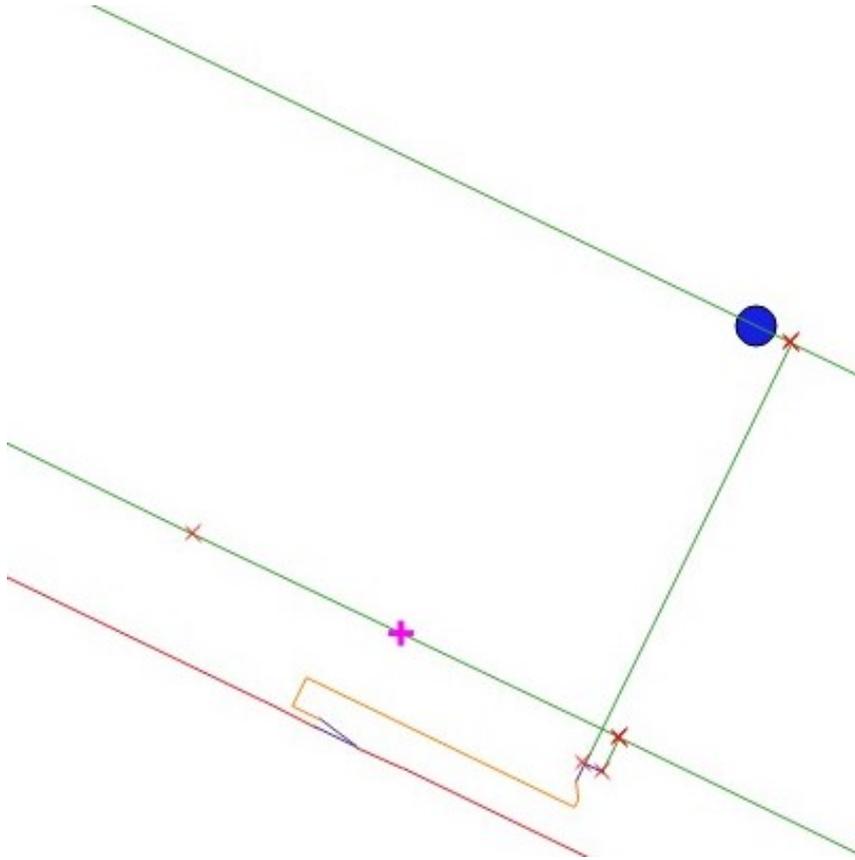


Figura 4.41: Step: 6 - Aggiunta elementi

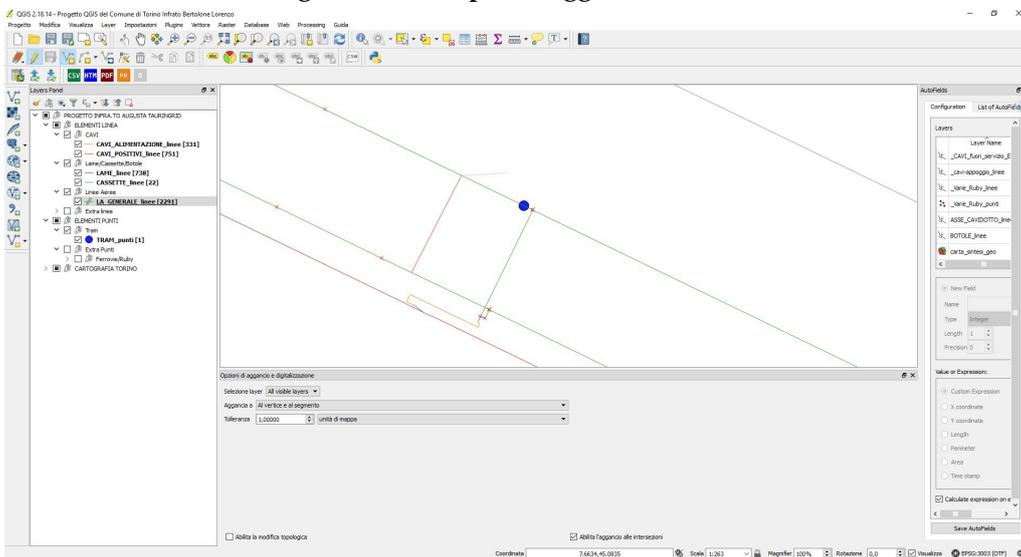


Figura 4.42: Step: 7 - Aggiunta elementi

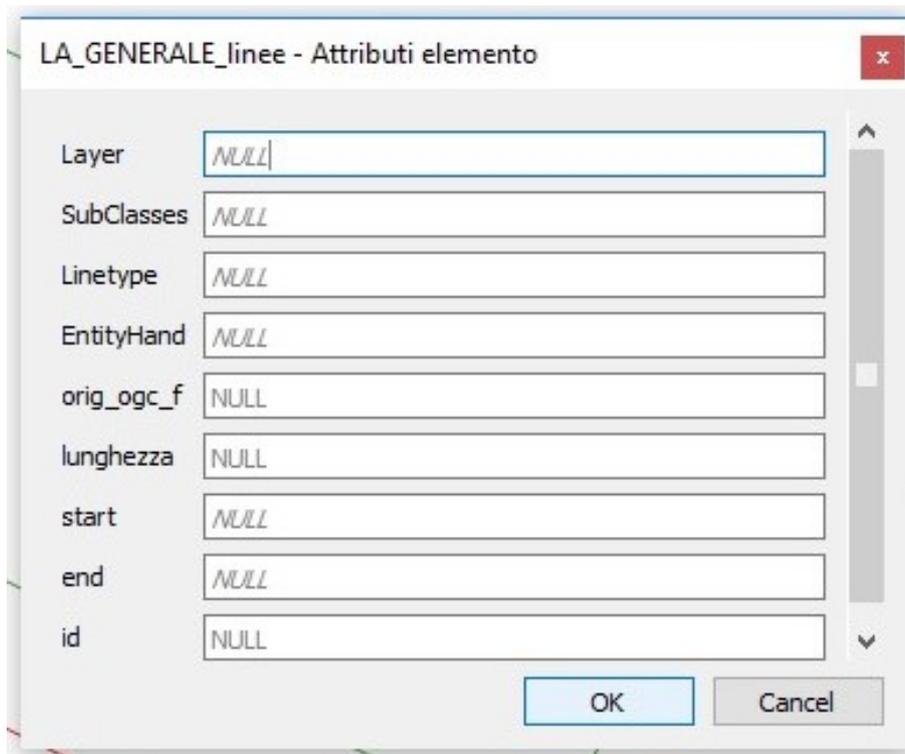


Figura 4.43: Step: 8 - Aggiunta elementi

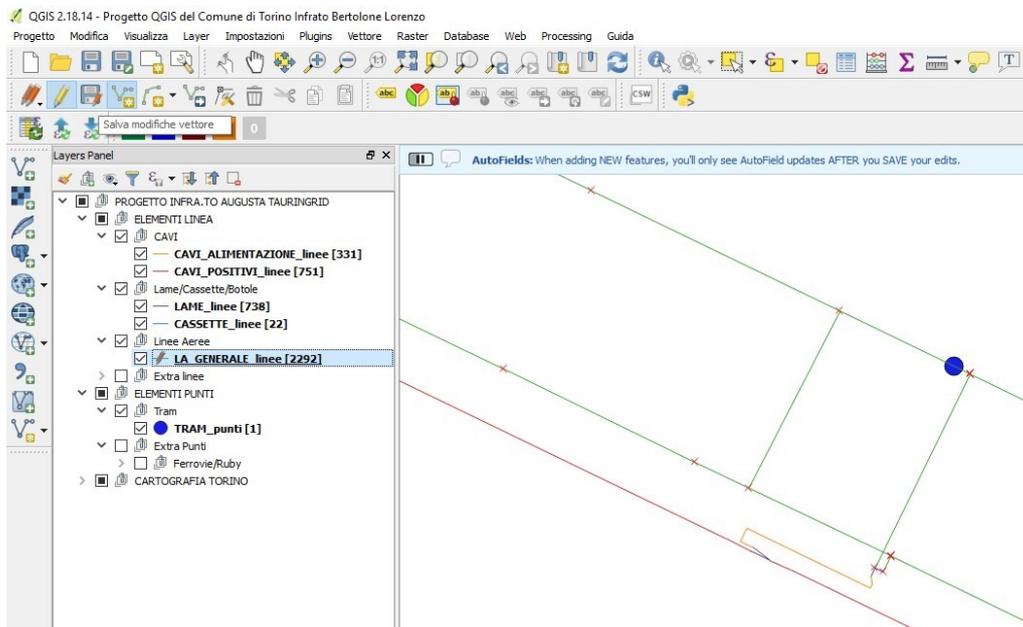


Figura 4.44: Step: 9 - Aggiunta elementi

4 – Progetto 'Augusta Tauringrid'

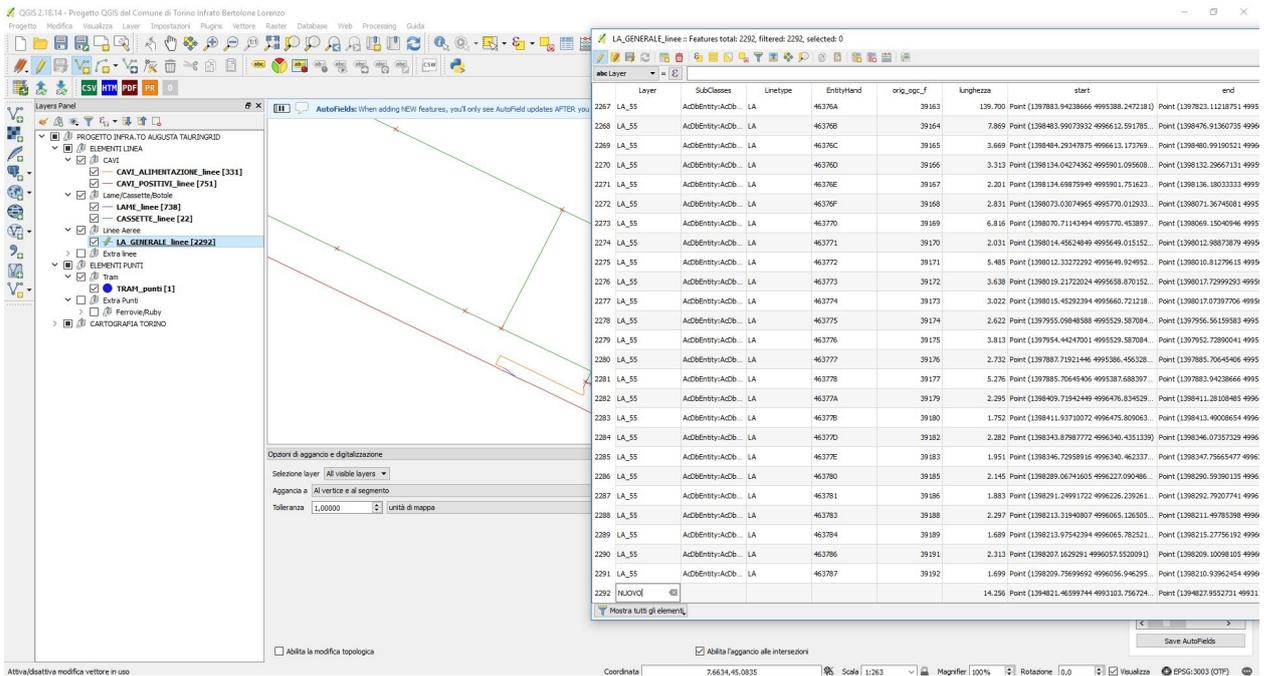


Figura 4.45: Step: 10 - Aggiunta elementi

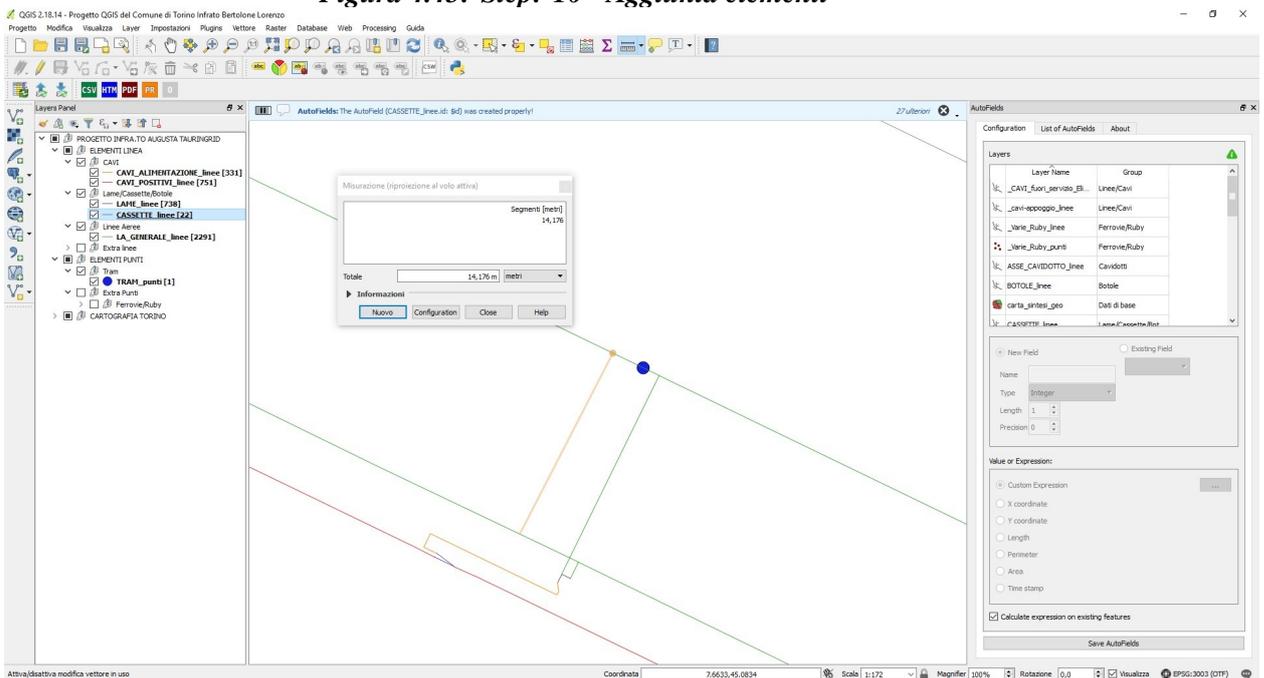


Figura 4.46: Misurazione entità

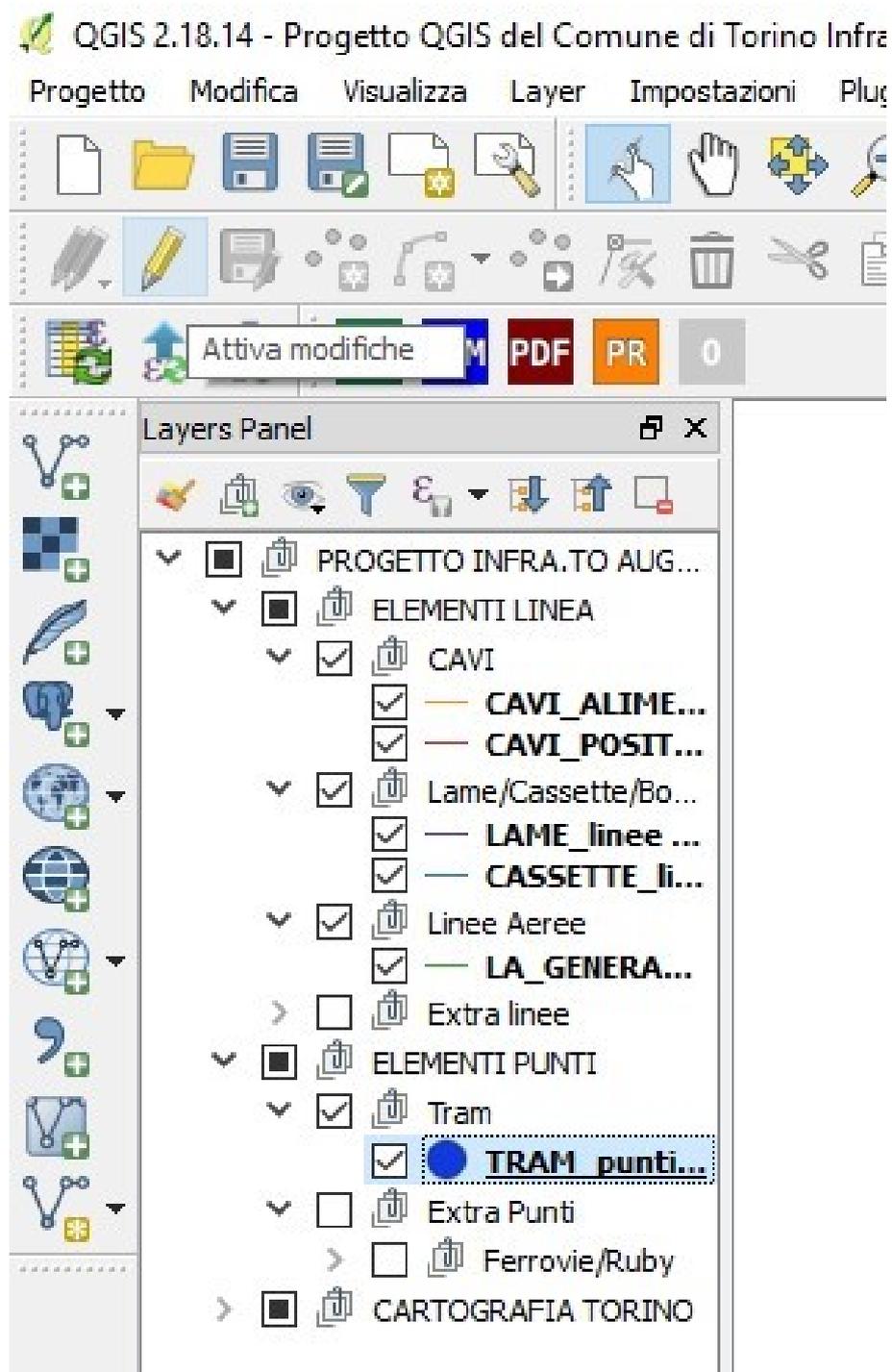


Figura 4.47: Step: 1 - Modifica elementi

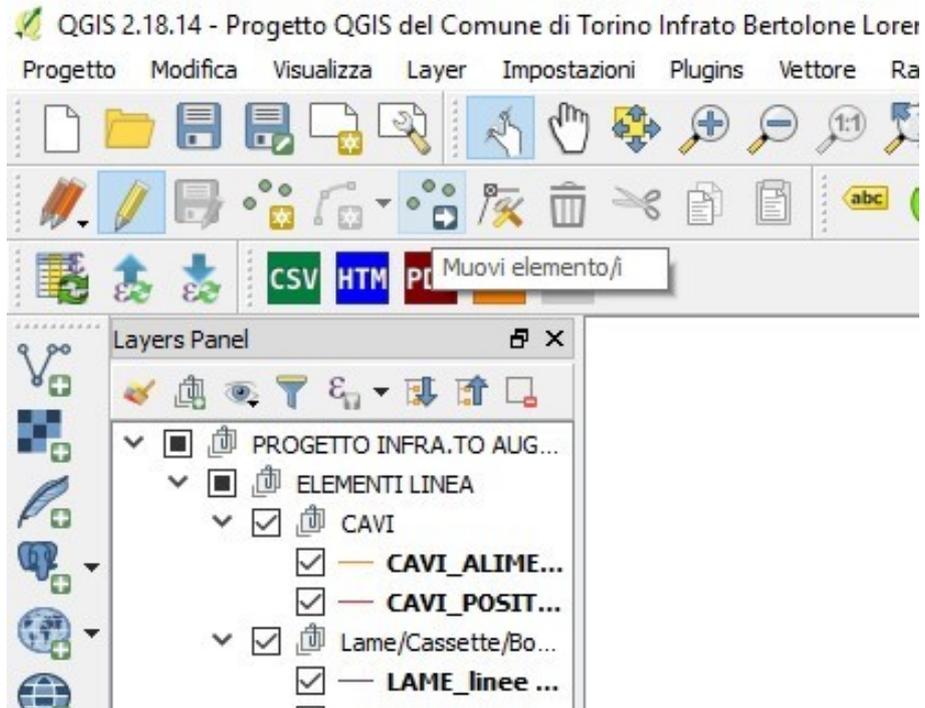


Figura 4.48: Step: 2 - Modifica elementi

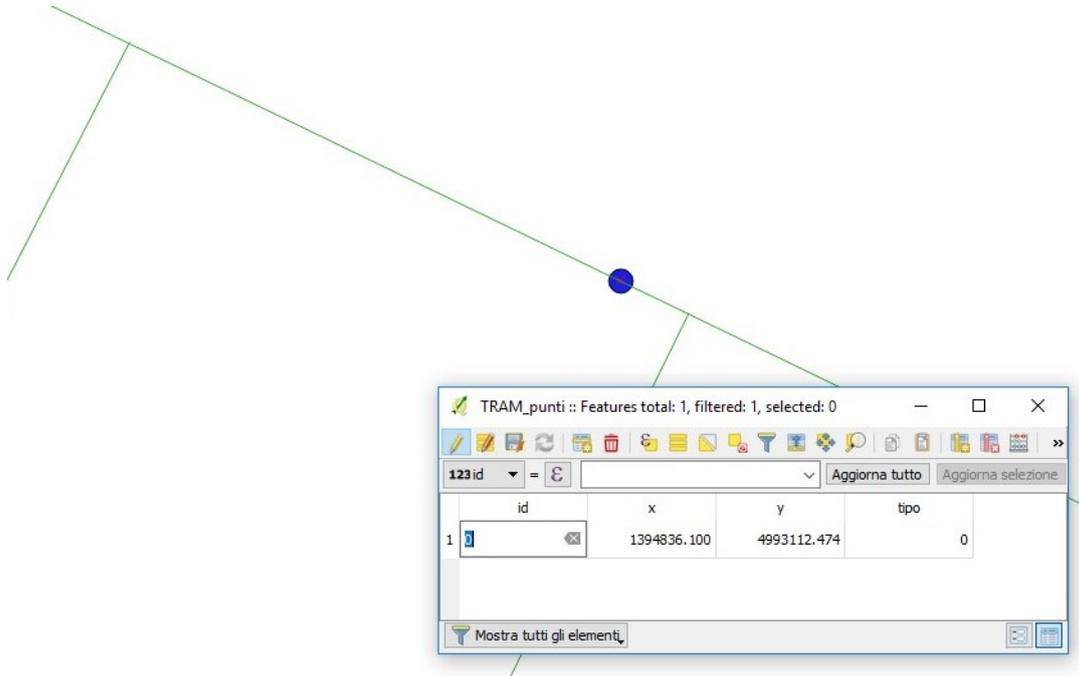


Figura 4.49: Step: 3 - Modifica elementi

4 – Progetto 'Augusta Tauringrid'

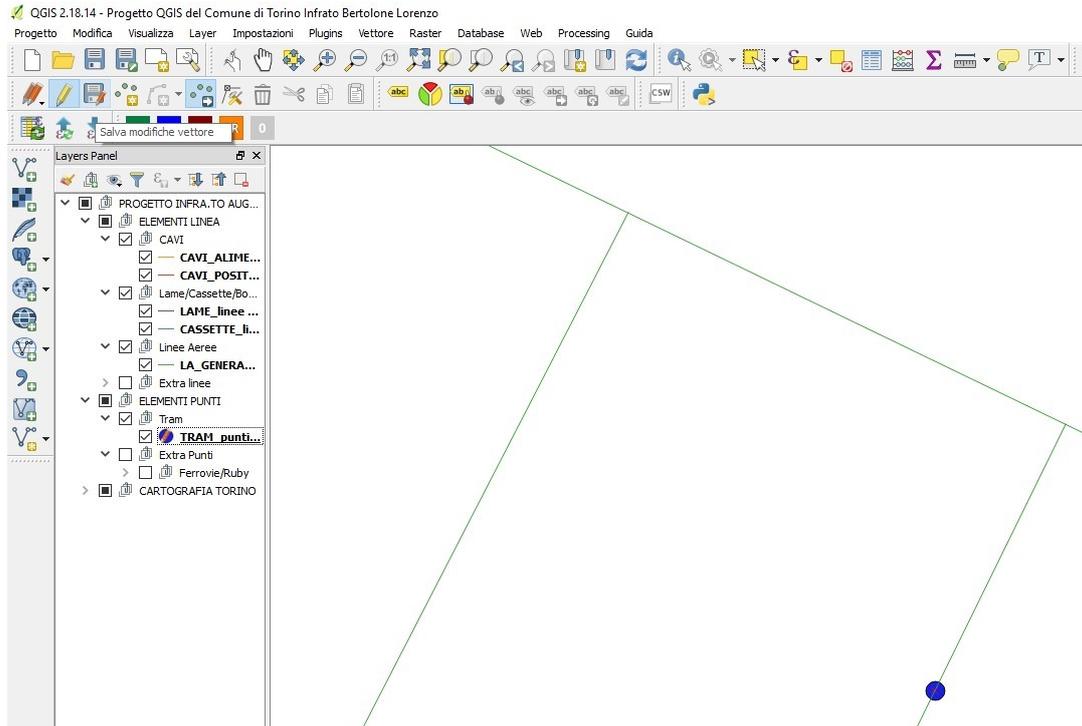


Figura 4.50: Step: 4 - Modifica elementi

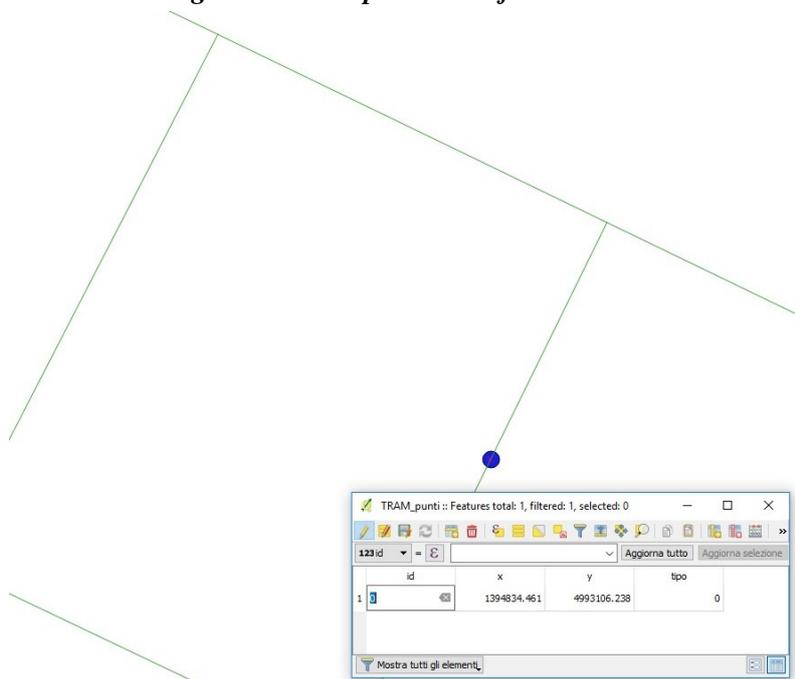


Figura 4.51: Step: 5 - Modifica elementi

4.4 Esportazione da ambiente QGIS

Una volta impostate tutte le modifiche e realizzata la rete è possibile esportare il contenuto tramite il plugin QuickExport in maniera molto semplice, affidabile e veloce. Tramite la barra apposita precedentemente illustrata facendo click su **CSV**, dopo aver selezionato il singolo layer da esportare, verrà data la possibilità di indirizzare una cartella di salvataggio predefinita e di effettuare il salvataggio con le convenzioni precedentemente illustrate affinché sia possibile l'importazione corretta in Matlab come illustrato da figura 4.52 a figura 4.54. Se per esempio si vuole analizzare esclusivamente una determinata zona della rete oppure determinati componenti è sufficiente modificare il formato salvato in .csv del layer corrispondente in maniera veloce ed estremamente affidabile.

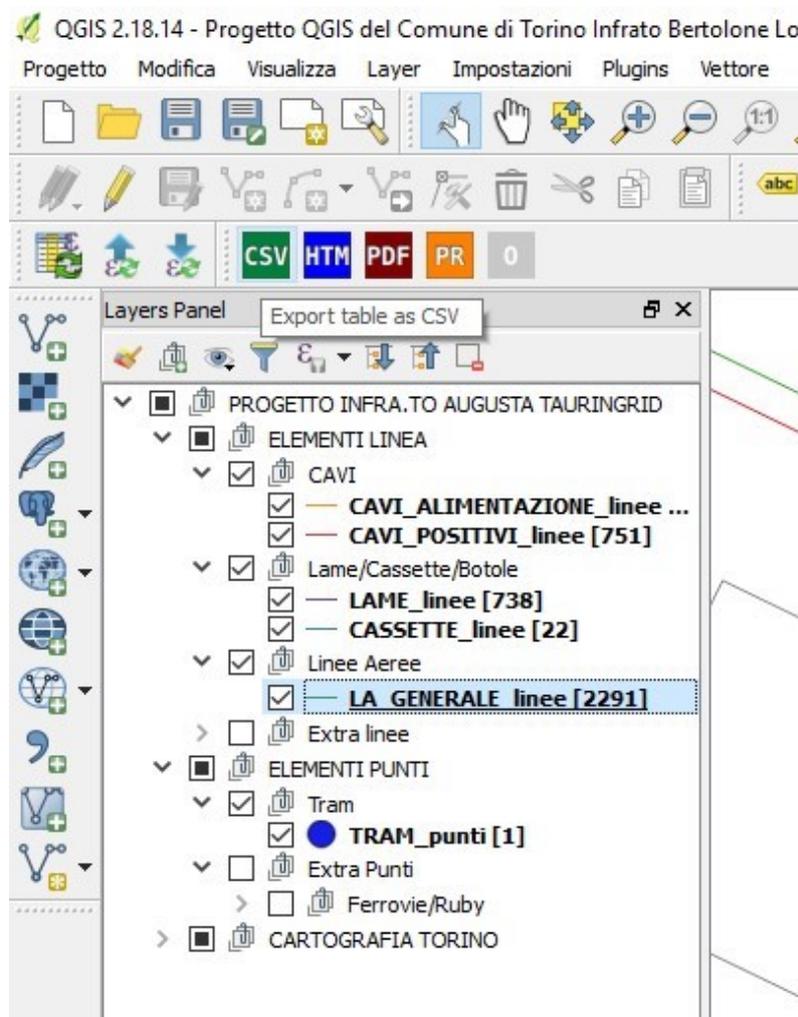


Figura 4.52: Fase: 1 - Esportazione .CSV

4 – Progetto 'Augusta Tauringrid'

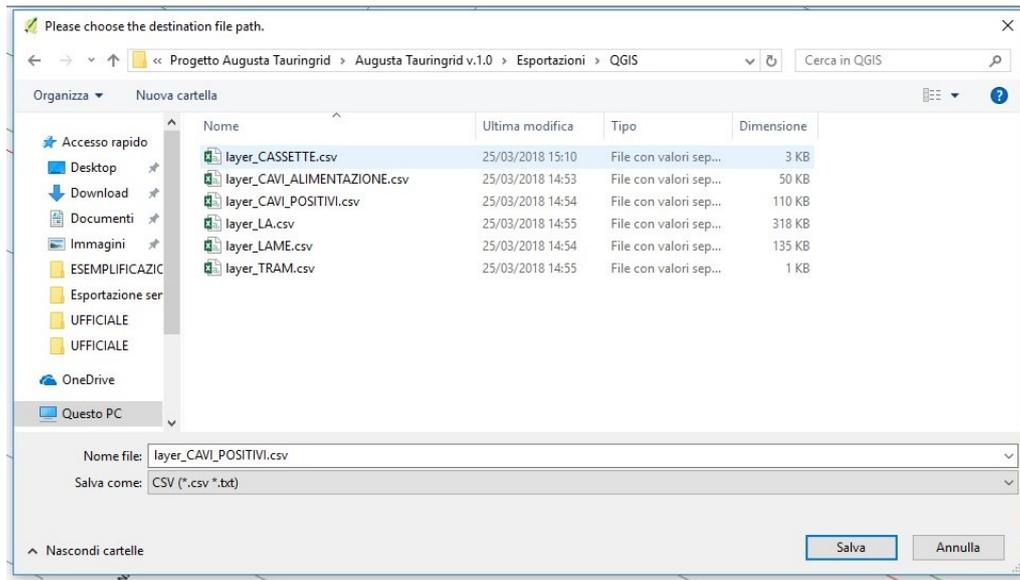


Figura 4.53: Fase: 2 - Esportazione .CSV

layer_CAVI_POSITIVI.csv - Excel

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
34	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,DB4,50,106.111,Point (1394044.02963748 4988551.09217438),Point (1394081.08754344 4988497.30312866),32															
35	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,DB8,51,185.572,Point (1394043.94060004 4988551.13769728),Point (1394152.93862263 4988460.56384243),33															
36	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,DBC,52,185.992,Point (1394043.85156261 4988551.18322018),Point (1394152.80303783 4988460.29622934),34															
37	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,DBD,53,186.296,Point (1394043.76252517 4988551.22874308),Point (1394152.66745304 4988460.02861626),35															
38	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,DBE,54,186.623,Point (1394043.67348773 4988551.27426599),Point (1394152.53186824 4988459.76100317),36															
39	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,DCT,55,1151.403,Point (1394044.74193699 4988550.72799117),Point (1394672.45879764 4989066.56618049),37															
40	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,E46,56,415.08,Point (1394043.58256003 4988551.31608304),Point (1393689.02574284 4988697.56777272),38															
41	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,E4B,57,435.235,Point (1394042.23276784 4988552.00208392),Point (1393670.01381391 4988700.58222349),39															
42	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,E4C,58,435.295,Point (1394042.01595198 4988552.11216581),Point (1393669.7467016 4988700.71879219),40															
43	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,E62,59,433.703,Point (1394043.22679014 4988551.49891636),Point (1393696.38468515 4988752.16054517),41															
44	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,E7A,60,405.537,Point (1394673.66656594 4989067.50495401),Point (1394921.09766376 4989377.14418837),42															
45	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,E85,61,348.489,Point (1394924.1809319 4989378.30187227),Point (1395139.69167085 4989620.81605136),43															
46	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,E87,62,338.006,Point (1395140.64233514 4989621.97632608),Point (1395351.61133187 4989879.29203728),44															
47	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,F51,65,80.18,Point (1394045.37698491 4988550.40573713),Point (1394076.52142639 4988499.26347936),45															
48	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F52,66,1304.326,Point (1394082.52737555 4988511.00980848),Point (1394615.67422774 4989617.49152775),46															
49	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5E,70,414.914,Point (1394615.40642624 4989617.62674002),Point (1394230.60492292 4989737.87391875),47															
50	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5F,71,415.642,Point (1394615.94202923 4989617.35631547),Point (1394230.45692622 4989737.61296475),48															
51	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5G,73,590.299,Point (1393622.01923968 4991418.31142111),Point (1394121.50616357 4991132.48649853),49															
52	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5H,74,589.899,Point (1393622.19476711 4991418.21555598),Point (1394121.65061169 4991132.7494334),50															
53	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5I,75,414.927,Point (1394122.82083789 4991131.76425797),Point (1394222.8293008 4990779.50163716),51															
54	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,F5J,76,415.391,Point (1394122.96528601 4991132.02719283),Point (1394222.1293008 4990779.50163716),52															
55	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1003,77,651.81,Point (1394221.76922258 4990786.12945543),Point (1393726.08293987 4991016.30662235),53															
56	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,13FF,104,220.274,Point (1395287.12317427 4988219.63652769),Point (1395226.79990625 4988016.25704899),54															
57	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1400,105,220.898,Point (1395287.98753299 4988219.38576087),Point (1395227.08802583 4988016.17346005),55															
58	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1402,106,358.133,Point (1395226.38196155 4988014.81645112),Point (1395130.21194589 4987677.18961533),56															
59	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1405,108,663.42,Point (1395131.6521558 4987676.77033567),Point (1394965.65828125 4987066.6999424),57															
60	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1406,109,663.886,Point (1395131.73601173 4987677.05837766),Point (1394965.54632323 4987066.61608647),58															
61	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1457,112,427.47,Point (1393594.28769438 4986594.69083832),Point (1393841.14572682 4986641.53481669),59															
62	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1458,113,426.949,Point (1393594.37653619 4986594.64493481),Point (1393840.58946076 4986641.75969019),60															
63	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,1458,114,601.171,Point (1393841.70791058 4986642.92548178),Point (1394054.97291883 4987195.46496927),61															
64	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,145C,115,601.181,Point (1393841.15164454 4986643.15035529),Point (1394054.41665279 4987195.68984277),62															
65	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14A4,116,340.927,Point (1393984.81665008 4987549.1306225),Point (1394057.62026217 4987836.24056998),63															
66	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14A8,117,1411.594,Point (1393984.63529295 4987549.21494089),Point (1393054.37362355 4987083.05591485),64															
67	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14A9,118,1411.009,Point (1393984.54461439 4987549.25710008),Point (1393054.56268255 4987083.28884551),65															
68	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14AB,119,1577.815,Point (1393984.45393582 4987549.29925928),Point (1392893.29847229 4987302.9887862),66															
69	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14EB,120,54.694,Point (1394045.10987542 4988550.54231137),Point (1394053.42661434 4988510.72524398),67															
70	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,1000,14EF,121,54.995,Point (1394045.02083893 4988550.58783612),Point (1394053.2905439 4988510.4578775),68															
71	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,14EB,122,963.461,Point (1394053.82050246 4988509.51493321),Point (1394057.07754571 4987836.49641924),69															
72	CAVI_POSITIVI,AcDbEntity:AcDbPolyline,0500,14EF,123,963.884,Point (1394053.55312588 4988509.65103651),Point (1394058.80618248 4987836.67343388),70															

Figura 4.54: Fase: 3 - Esportazione .CSV

4.5 Importazione in ambiente MATLAB

Una volta esportati tutti i dati della rete è necessario accedere all'ambiente Matlab da cui sarà necessario avviare la procedura di run dello script. Una volta avviata l'elaborazione verrà esportato in maniera automatica un file sempre in formato *.csv* ed uno shapefile *.shp* per l'importazione in QGIS e l'elaborazione grafica. Si rammenta che è possibile procedere in due modi differenti: il primo metodo consiste nel salvare direttamente lo shapefile originale sulla cartella di lavoro di Matlab mentre la seconda è quella di fare una copia degli shapefile dalla cartella sorgente originale del progetto ogni volta si necessita di strutturare una nuova simulazione. Comunque tutti i dettagli verranno illustrati nel capitolo inerente la simulazione del software per comprenderne vantaggi e svantaggi in funzione delle esigenze del committente. E' necessario interfacciarsi con un file *.csv* che verrà generato in funzione di una flag posta nelle variabili globali del codice per l'aggiornamento di tutte le connessioni relative tra le cabine connesse alla rete. Viene elencata la procedura corretta per la realizzazione della tabella aggiornata delle cassette ogni qualvolta ne venga aggiunta una o strutturata una differente. In sintesi, se per la prima volta si è modificata la struttura della rete e nello specifico si è aggiunta o modificata una o più cassette o cabine è sufficiente agire sulle variabili globali di `print` nel codice *BASE* e porre ad 1 la variabile `print`. Una volta eseguita questa operazione ed avviato il codice *BASE* verrà automaticamente eseguita la procedura di *TEST*. Una volta generato il file *.csv* contenente la lista aggiornata delle cassette è possibile modificarne l'interfaccia manualmente aggiungendo eventuali note laterali per associare connessioni ad identificazioni precise della rete e salvare sulla stessa cartella il risultato finale come *.csv*. Se la rete è sempre la medesima per numero di cabine o cassette è possibile evitare il riaggiornamento della lista ponendo la variabile globale `print` pari a zero.

1. Si elenca la procedura da figura 4.55 a figura 4.65 nel caso si debbano aggiornare le connessioni di cabina. Se si vuole effettuare la procedura di aggiornamento delle cabine bisognerà richiamare la flag dalle variabili globali e verrà automaticamente realizzato un *.csv* che risulterà essere già di suo modificabile e semplificabile con un'interfaccia per l'aggiunta di eventuali annotazioni per l'identificazione di ogni lato connesso alla cabina. Il salvataggio dell'interfaccia in formato *.csv* è fondamentale affinché il codice possa associare uno stato di interruttore aperto o chiuso per ogni lato connesso a ciascuna cabina della rete.

4 – Progetto 'Augusta Tauringrid'

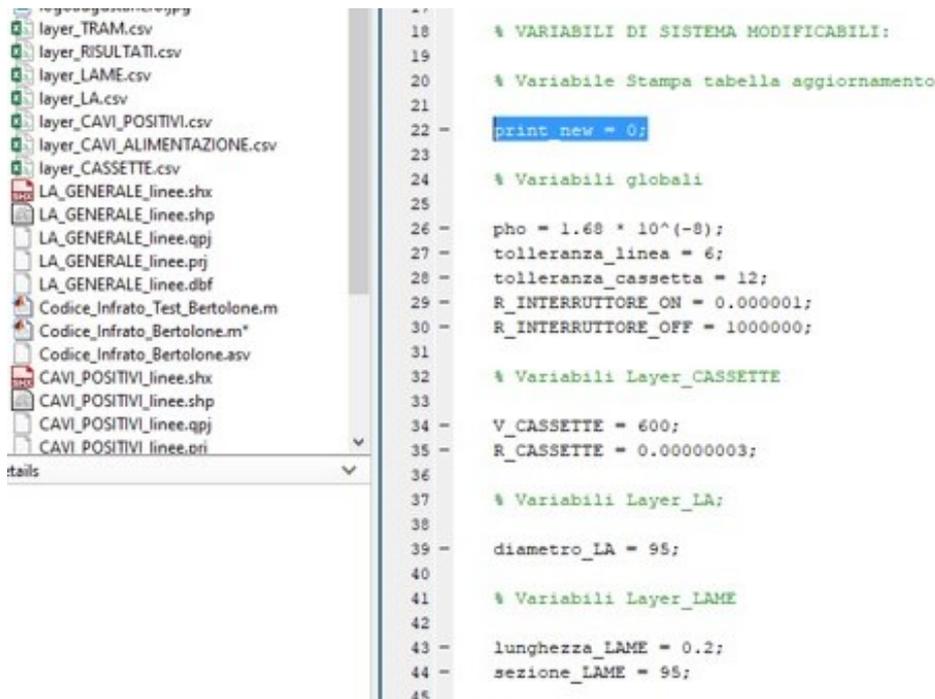


Figura 4.55: Step: 1 - Interfaccia refresh

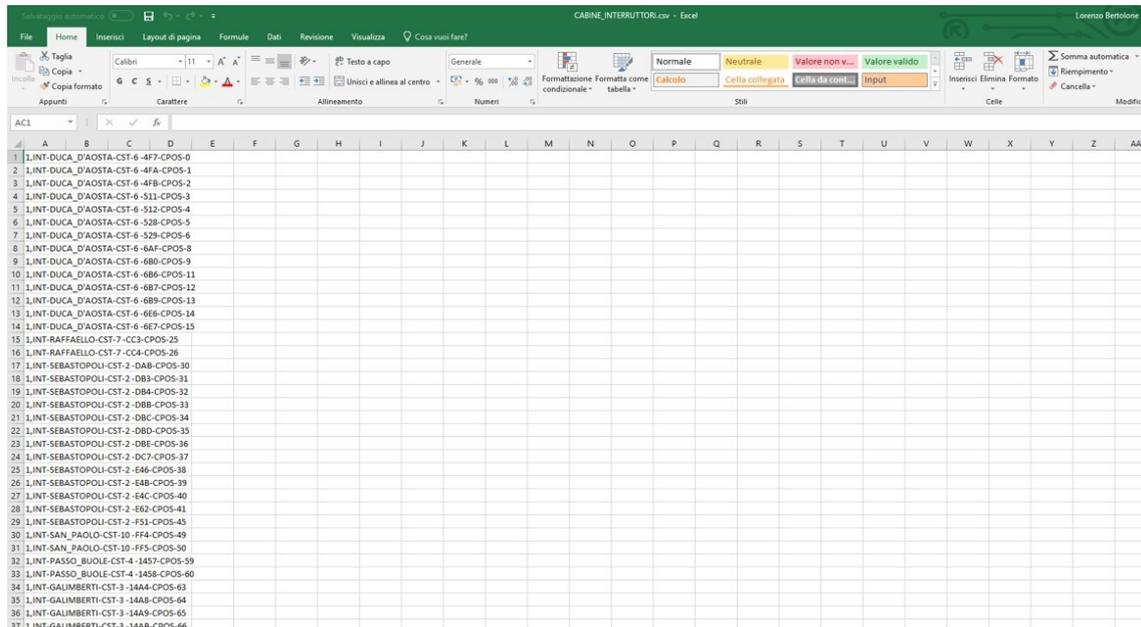


Figura 4.56: Step: 2 - Interfaccia refresh

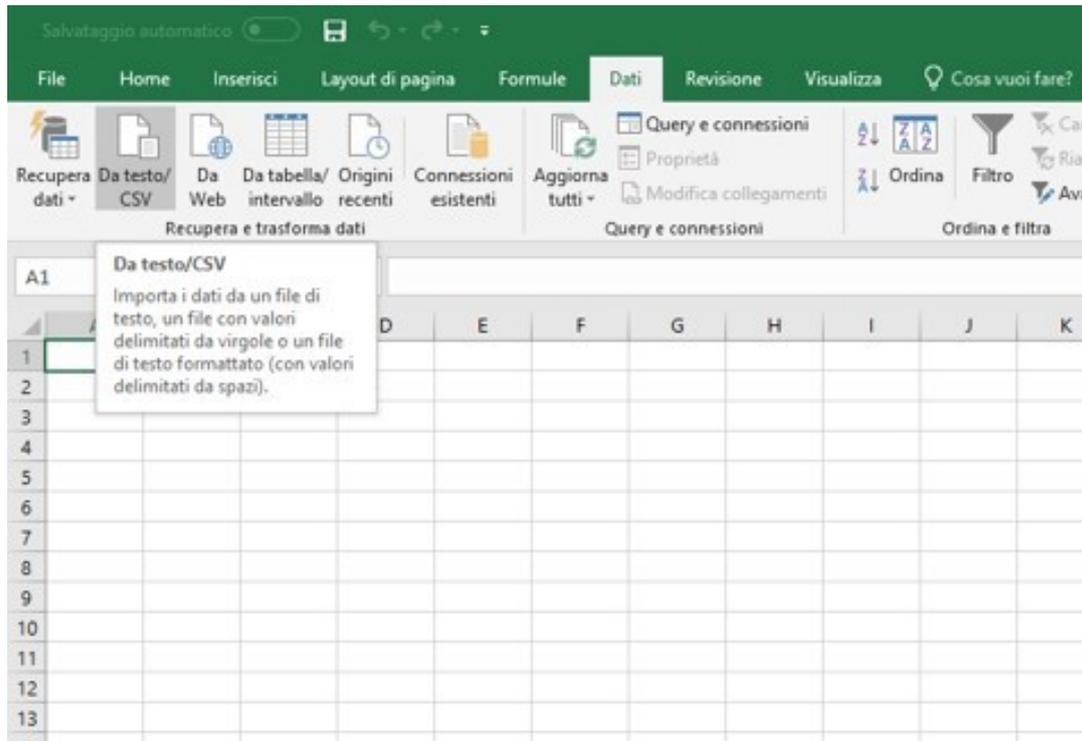


Figura 4.57: Step 3 - Interfaccia refresh

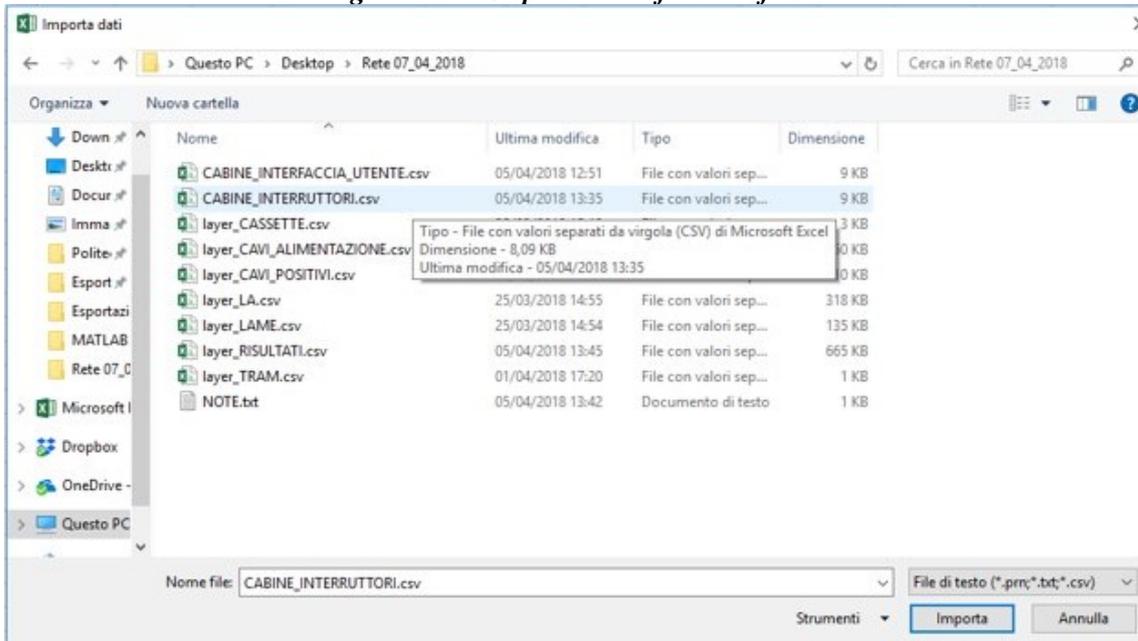


Figura 4.58: Step 4 - Interfaccia refresh

4 – Progetto 'Augusta Tauringrid'

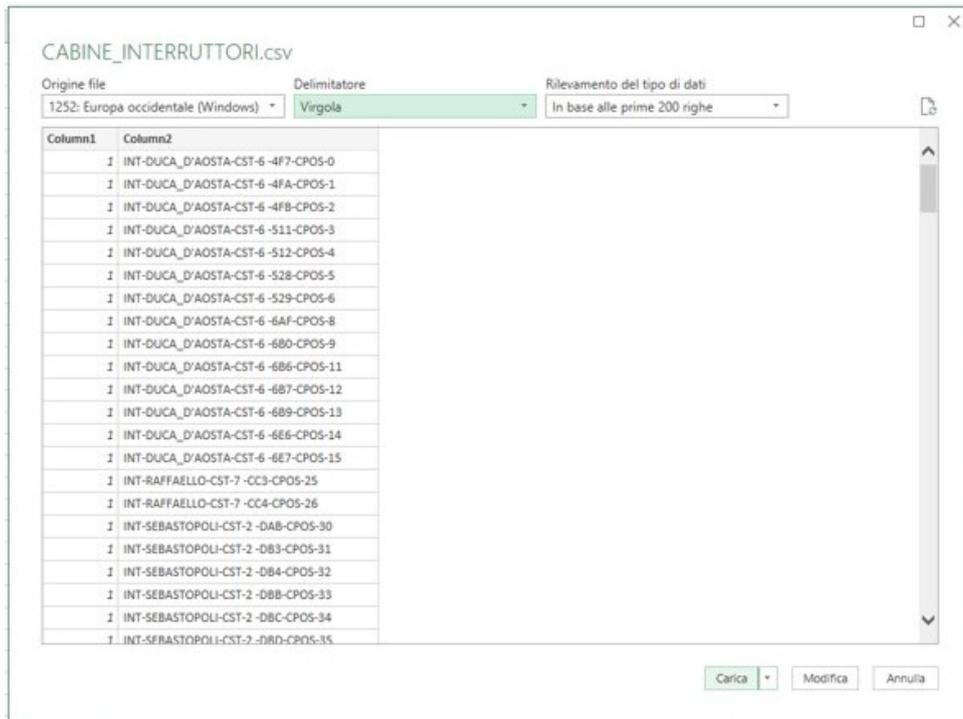


Figura 4.59: Step: 5 - Interfaccia refresh

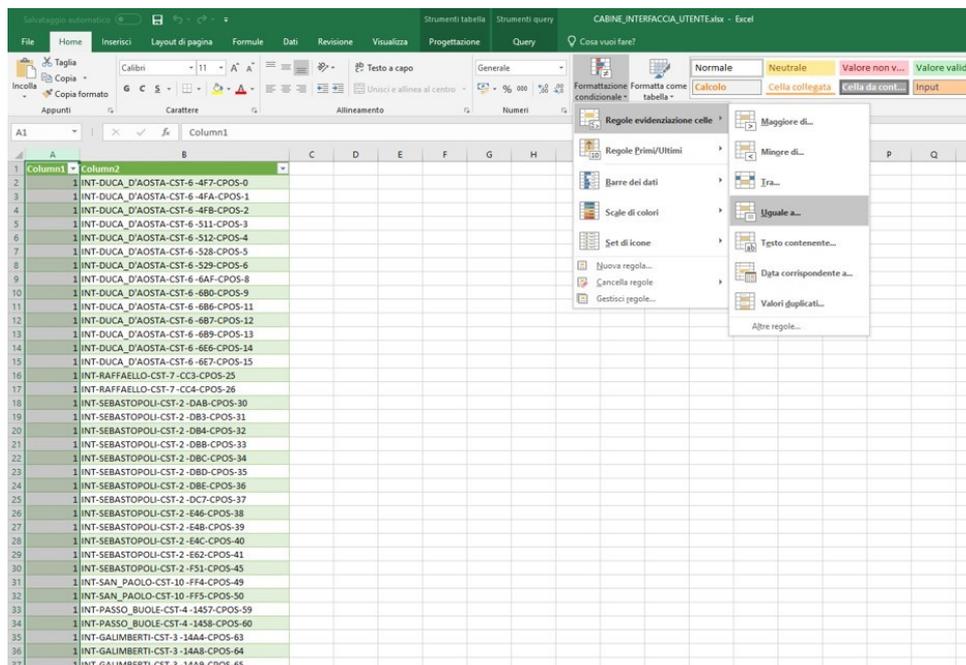


Figura 4.60: Step: 6 - Interfaccia refresh

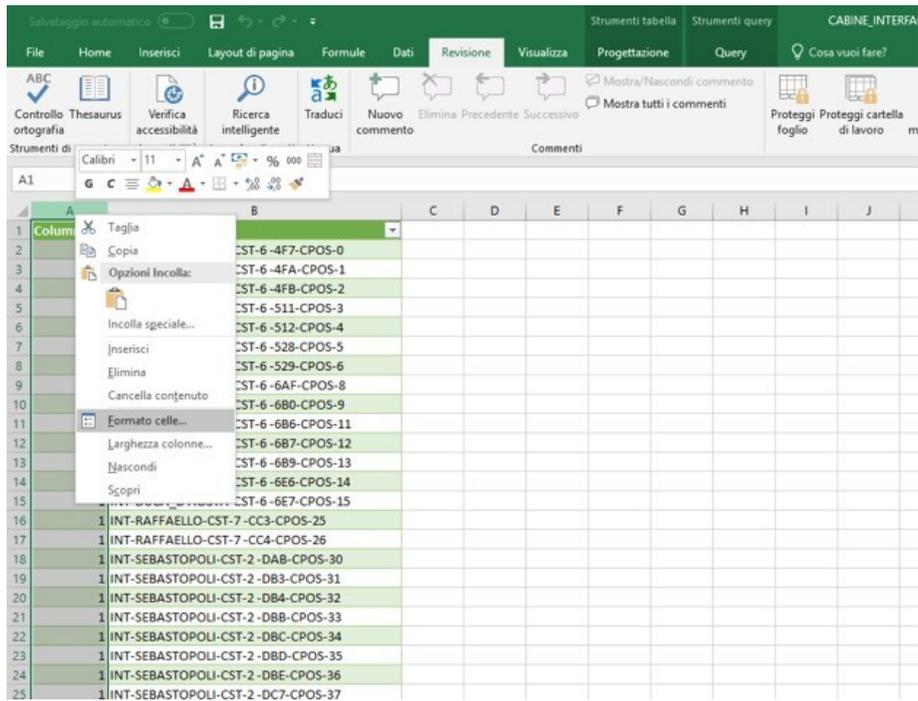


Figura 4.61: Step: 7 - Interfaccia refresh

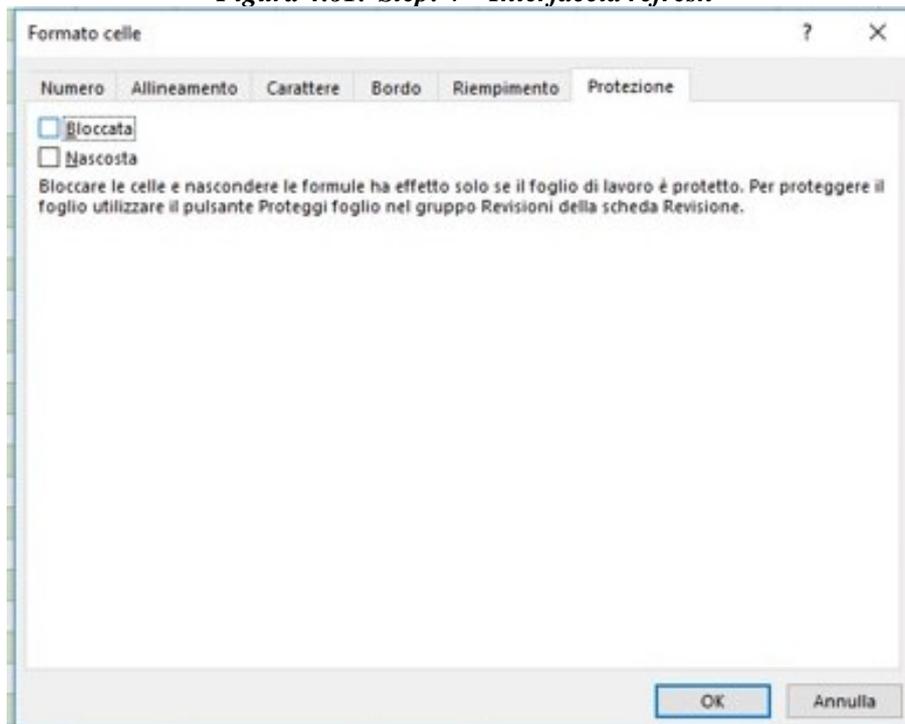


Figura 4.62: Step: 8 - Interfaccia refresh

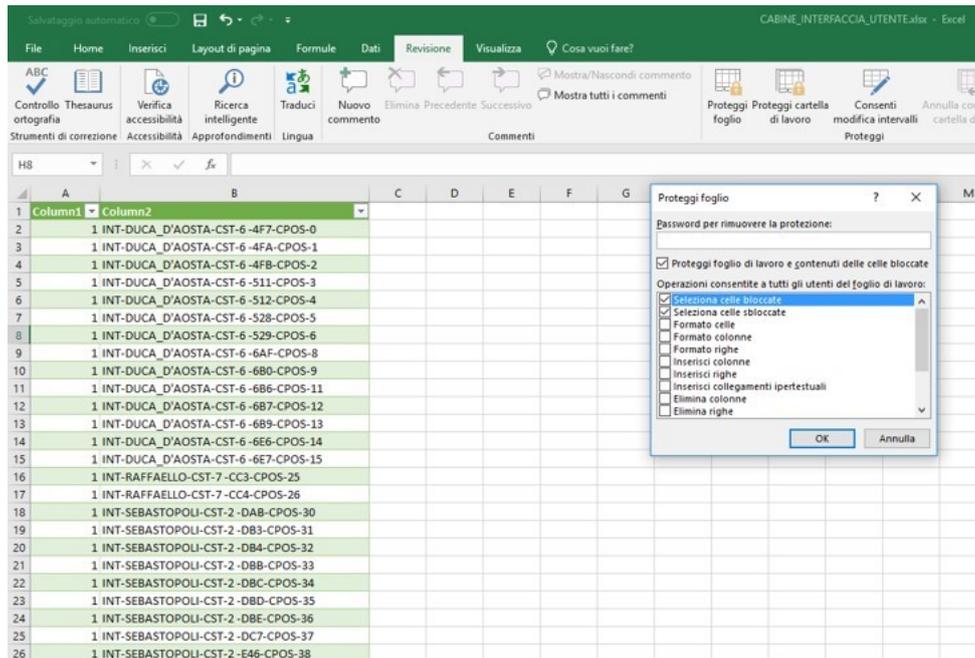


Figura 4.63: Step: 9 - Interfaccia refresh

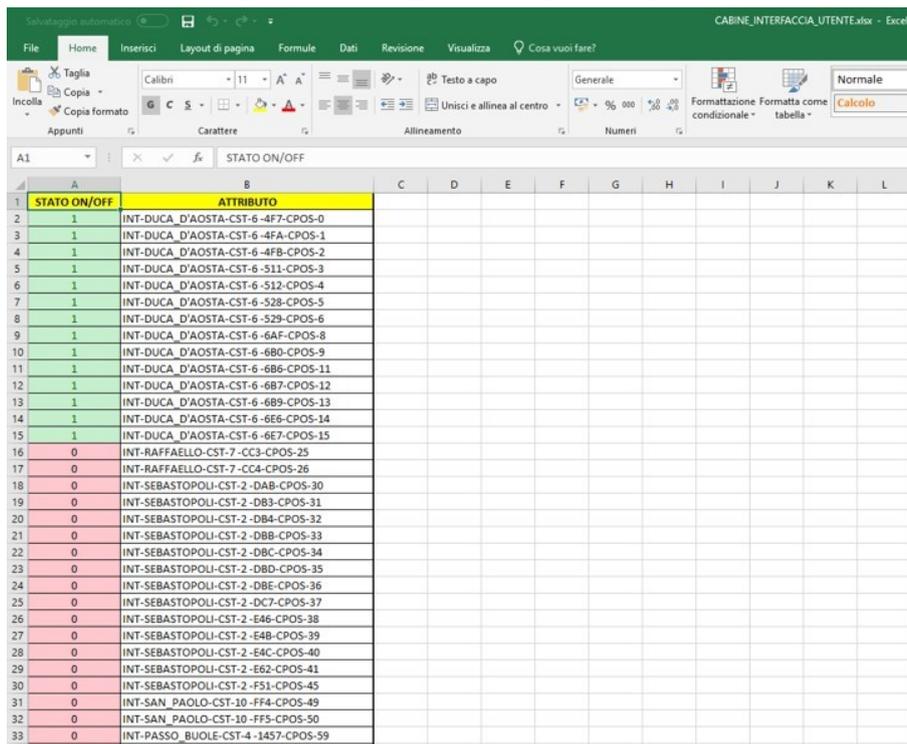
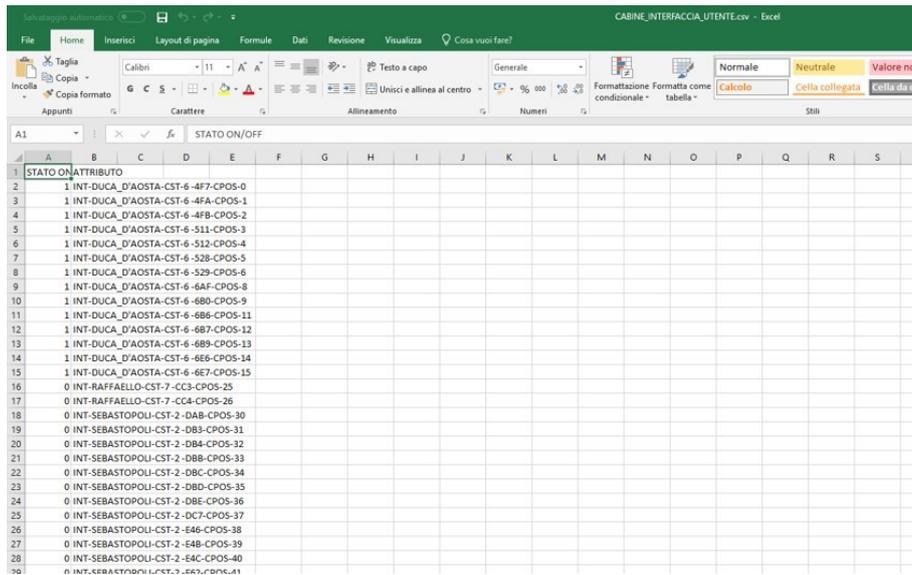


Figura 4.64: Step: 10 - Interfaccia refresh

4 – Progetto 'Augusta Tauringrid'



The screenshot shows an Excel spreadsheet with the following data:

A1	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	
1	STATO ON	ATTRIBUTO																		
2	1	INT-DUCA_D'AOSTA-CST-6-4F7-CPOS-0																		
3	1	INT-DUCA_D'AOSTA-CST-6-4FA-CPOS-1																		
4	1	INT-DUCA_D'AOSTA-CST-6-4FB-CPOS-2																		
5	1	INT-DUCA_D'AOSTA-CST-6-511-CPOS-3																		
6	1	INT-DUCA_D'AOSTA-CST-6-512-CPOS-4																		
7	1	INT-DUCA_D'AOSTA-CST-6-528-CPOS-5																		
8	1	INT-DUCA_D'AOSTA-CST-6-529-CPOS-6																		
9	1	INT-DUCA_D'AOSTA-CST-6-6AF-CPOS-8																		
10	1	INT-DUCA_D'AOSTA-CST-6-680-CPOS-9																		
11	1	INT-DUCA_D'AOSTA-CST-6-686-CPOS-11																		
12	1	INT-DUCA_D'AOSTA-CST-6-687-CPOS-12																		
13	1	INT-DUCA_D'AOSTA-CST-6-689-CPOS-13																		
14	1	INT-DUCA_D'AOSTA-CST-6-6E6-CPOS-14																		
15	1	INT-DUCA_D'AOSTA-CST-6-6E7-CPOS-15																		
16	0	INT-RAFFAELLO-CST-7-CC3-CPOS-25																		
17	0	INT-RAFFAELLO-CST-7-CC4-CPOS-26																		
18	0	INT-SEBASTOPOLI-CST-2-DAB-CPOS-30																		
19	0	INT-SEBASTOPOLI-CST-2-DB3-CPOS-31																		
20	0	INT-SEBASTOPOLI-CST-2-DB4-CPOS-32																		
21	0	INT-SEBASTOPOLI-CST-2-DB8-CPOS-33																		
22	0	INT-SEBASTOPOLI-CST-2-DBC-CPOS-34																		
23	0	INT-SEBASTOPOLI-CST-2-DBD-CPOS-35																		
24	0	INT-SEBASTOPOLI-CST-2-DBE-CPOS-36																		
25	0	INT-SEBASTOPOLI-CST-2-DC7-CPOS-37																		
26	0	INT-SEBASTOPOLI-CST-2-E46-CPOS-38																		
27	0	INT-SEBASTOPOLI-CST-2-E48-CPOS-39																		
28	0	INT-SEBASTOPOLI-CST-2-E4C-CPOS-40																		
29	0	INT-SEBASTOPOLI-CST-2-E47-CPOS-41																		

Figura 4.65: Step: 11 - Interfaccia refresh

2. Successivamente all'elaborazione del programma sarà visualizzata sulla *command window* e sul *workspace* una serie di dati oltre alla creazione di un nuovo file *.csv* contenente i risultati nella cartella di lavoro Matlab. Ogni qualvolta venga cliccato il tasto di run, i dati vengono cancellati e riaggiornati. Una volta avviato il run illustrato da figura 4.66 a figura 4.68 e consultato i dati sarà possibile ritornare in ambiente QGIS per una elaborazione grafica dei risultati di simulazione.

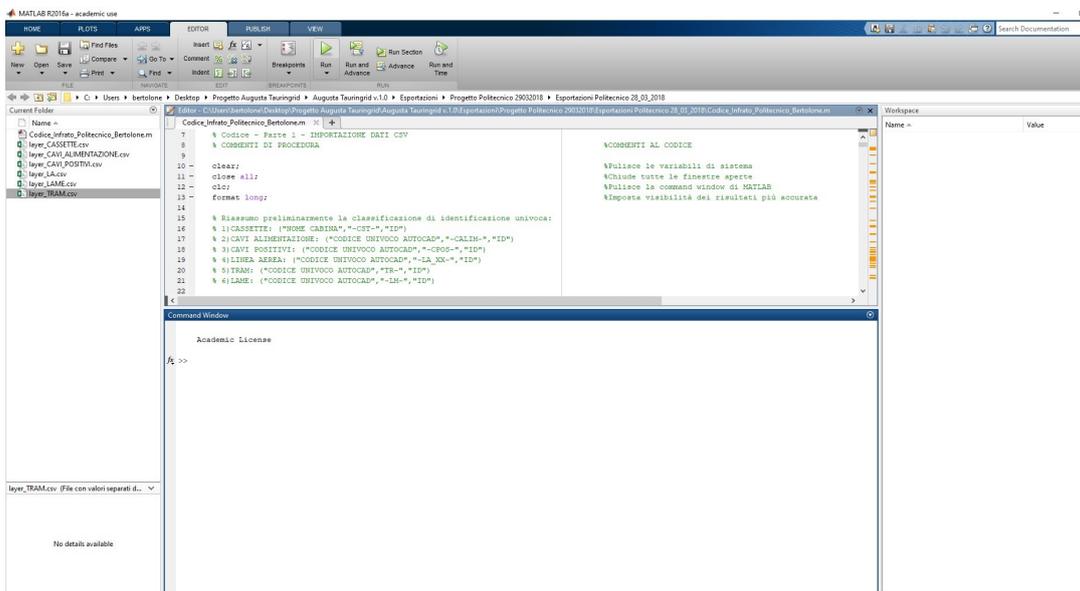


Figura 4.66: Step: 1 - Run Augusta Tauringrid

4 – Progetto 'Augusta Tauringrid'

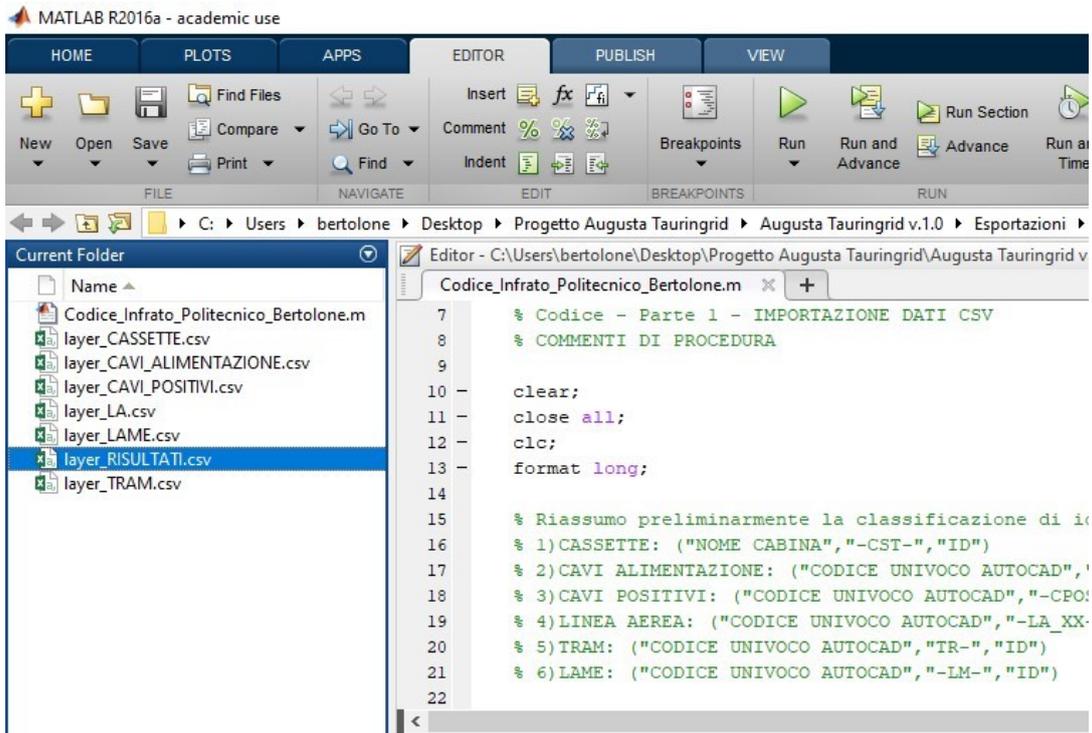


Figura 4.67: Step: 2 - Run Augusta Tauringrid

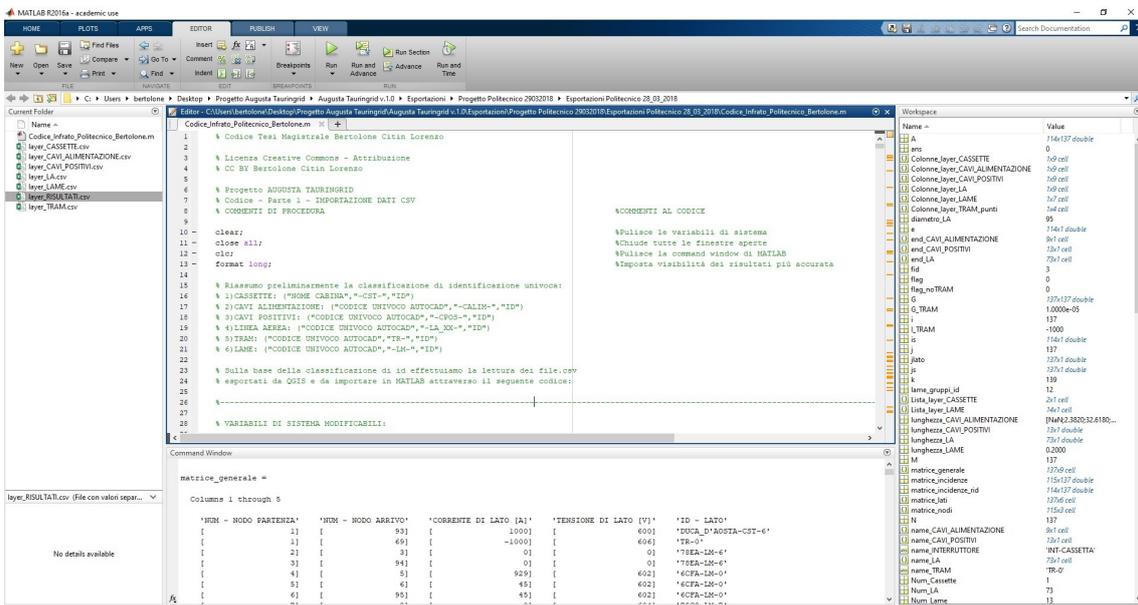


Figura 4.68: Step: 3 - Run Augusta Tauringrid

4.6 Elaborazione dati in ambiente MATLAB

Per completezza verrà illustrato il codice e la sua funzionalità. Il codice scritto in ambiente Matlab è suddiviso in 3 fasi. La prima fase è quella di lettura dei dati .csv esportati da QGIS ed importazione in Matlab codificandone attributi e proprietà geometriche degli elementi. La fase successiva è la ricostruzione della rete mediante algoritmi ulteriormente e successivamente migliorabili ed implementabili per il riconoscimento del tram sulla linea aerea o per esempio l'analisi delle cabine allacciate o meno alla rete. Lo scopo della seconda fase del codice è l'analisi dei dati importati per la ricostruzione della rete. Infine la fase finale è la procedura di calcolo automatico della rete [4] dove ho un'esportazione dei risultati in formato .csv e .shp che mi consentirà una visualizzazione più idonea in ambiente QGIS. Il logo del software è visibile in figura 4.69:

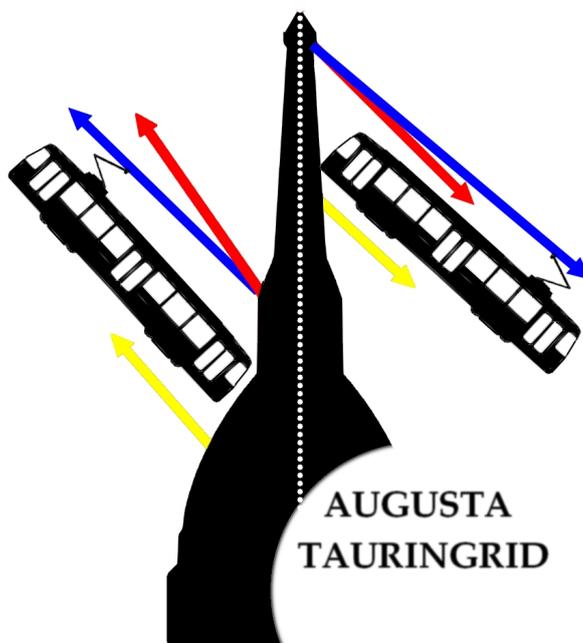


Figura 4.69: Logo Software Augustus Tauringrid Bertolone Citin Lorenzo

Si riporta lo schema esemplificato a blocchi in figura 4.70 per una comprensione delle principali funzionalità del codice e sarà premura successiva illustrare le parti più significative del codice che per sua interezza è reperibile nell'appendice della tesi. Si è deciso di suddividere il codice in 2 parti rispettivamente **BASE** e **TEST**. La prima parte permette di analizzare la struttura della rete e una volta avviato se non si modifica la rete strutturalmente aggiungendo cavi o linee non è necessario eseguirlo. La seconda parte di codice invece si occupa di modellizzare lame e tram. Questa suddivisione è stata resa preliminarmente necessaria per la notevole riduzione dei tempi di elaborazione del **software Augustus Tauringrid**. Quindi se bisogna effettuare più simulazioni variando la

configurazione delle lame o la posizione aggiunta di tram risulta strettamente necessario avviare la parte di codice di **TEST** sfruttando le *variabili pre-caricate* del codice **BASE** già avviato e concluso in fase preliminare. Al committente sulla base del file rilasciato all'incontro organizzato viene strutturata completamente la rete nelle 2 parti di codice senza necessità di eseguire il tutto da capo. Si riassume il funzionamento del codice suddividendo la rete I ovvero i cavi positivi, di alimentazione, le linee aeree dalla rete II che comprende la configurazione delle lame e dei tram.

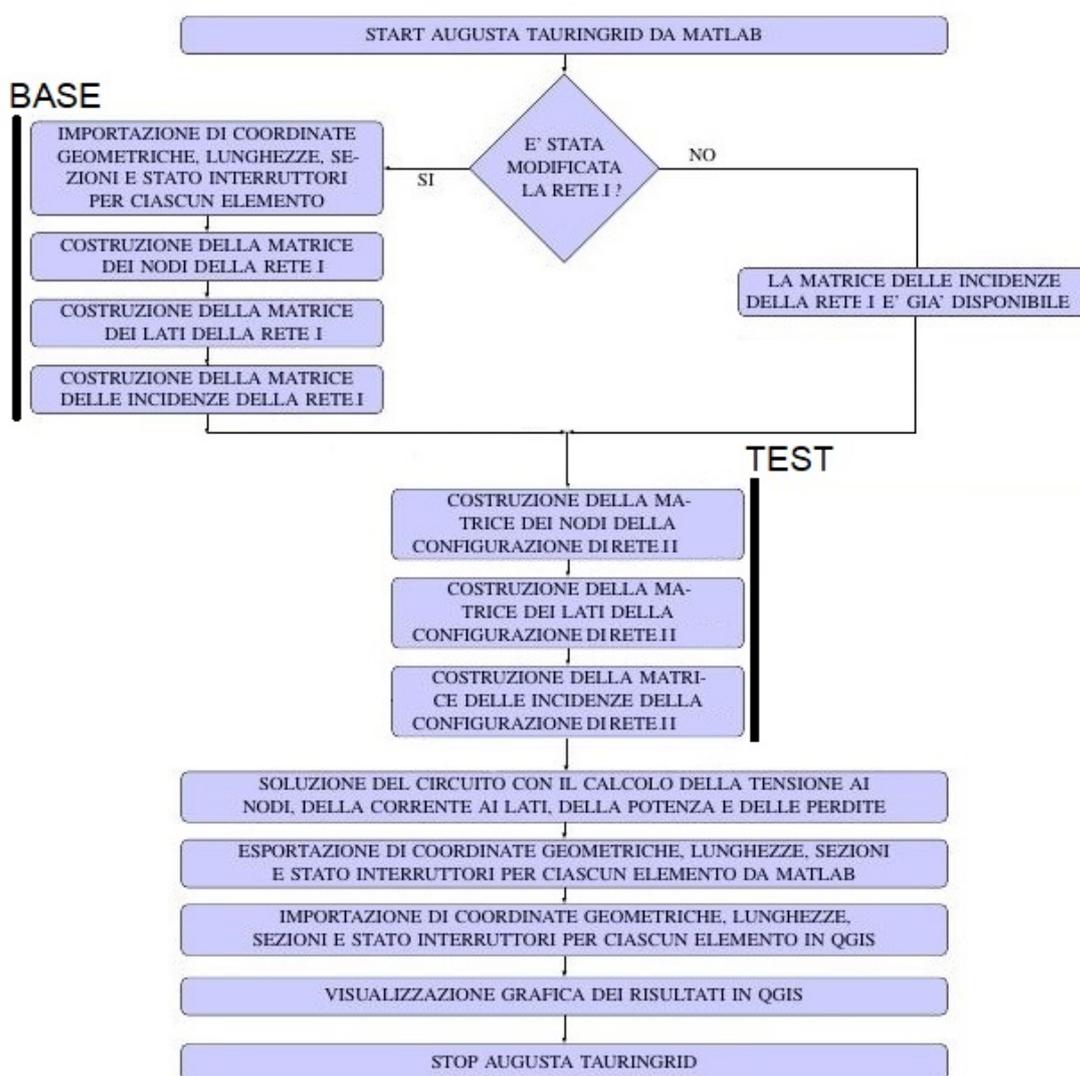


Figura 4.70: Schema a blocchi logica di funzionamento software Augusta Tauringrid

4.7 Visualizzazione dei Risultati su QGIS

Una volta esportati i risultati in Matlab bisogna importarli in QGIS seguendo la procedura successivamente illustrata da figura 4.71 a figura 4.73. La rappresentazione in QGIS permette una comprensione dei dati molto più immediata rispetto a Matlab. Consente di poter memorizzare più scenari divisi in sottogruppi. Nel capitolo successivo sulla base della configurazione del software appena illustrata verranno eseguiti i test di rete partendo dal caso più semplice a quello più complesso. Sulla base degli shapefile elaborati da Matlab è possibile importarli in QGIS tramite i seguenti passaggi commentati:

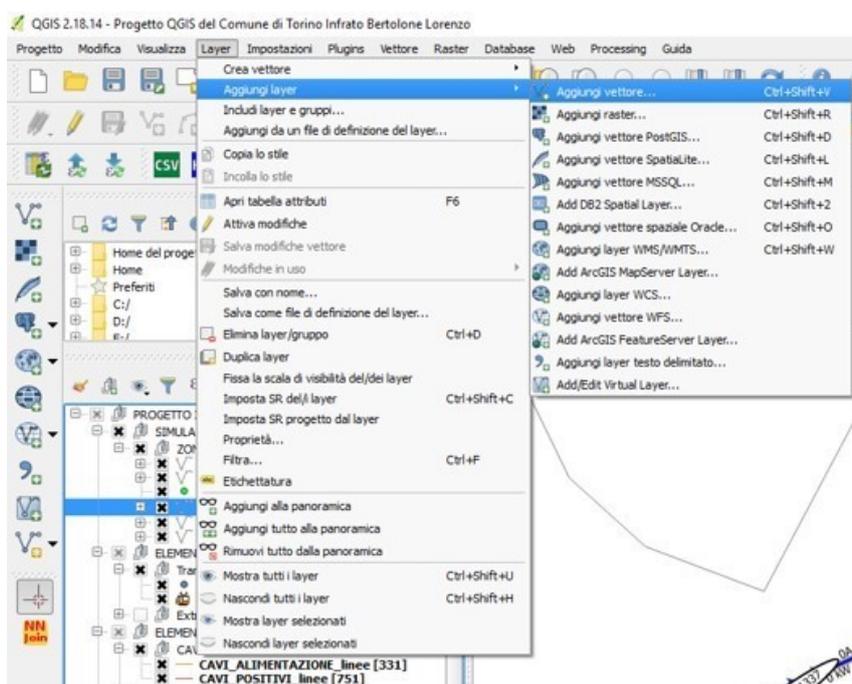


Figura 4.71: Fase: 1 - Importazione grafica risultati su QGIS

1. Si può caratterizzare un vettore inserito come shapefile con impostazioni predefinite quali: colore, simboli, etichette in funzione di formule complesse oppure di condizioni imposte. Si illustrano a grandi linee i vantaggi intrinseci per la visualizzazione in QGIS specificando che al committente viene rilasciata nella versione aggiornata un pacchetto di file salvati pre-impostati contenenti le specifiche già salvate. Nelle prossime illustrazioni da figura 4.74 a figura 4.90 viene illustrata la procedura per l'impostazione della visualizzazione grafica degli attributi di corrente, potenza e perdite in funzione del calcolo automatico eseguito. La procedura a grandi linee è rispecchiata per ogni vettore di interesse.

4 – Progetto 'Augusta Tauringrid'

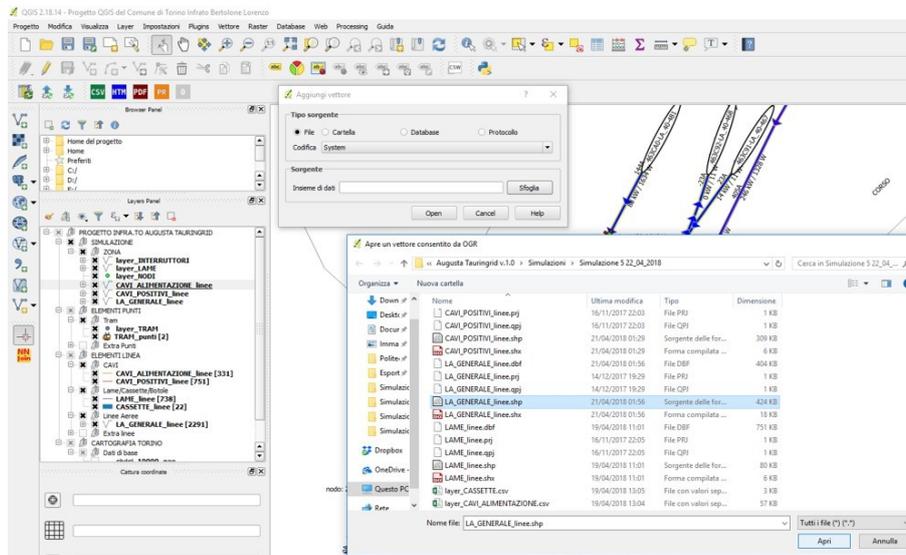


Figura 4.72: Fase: 2 - Importazione grafica risultati su QGIS

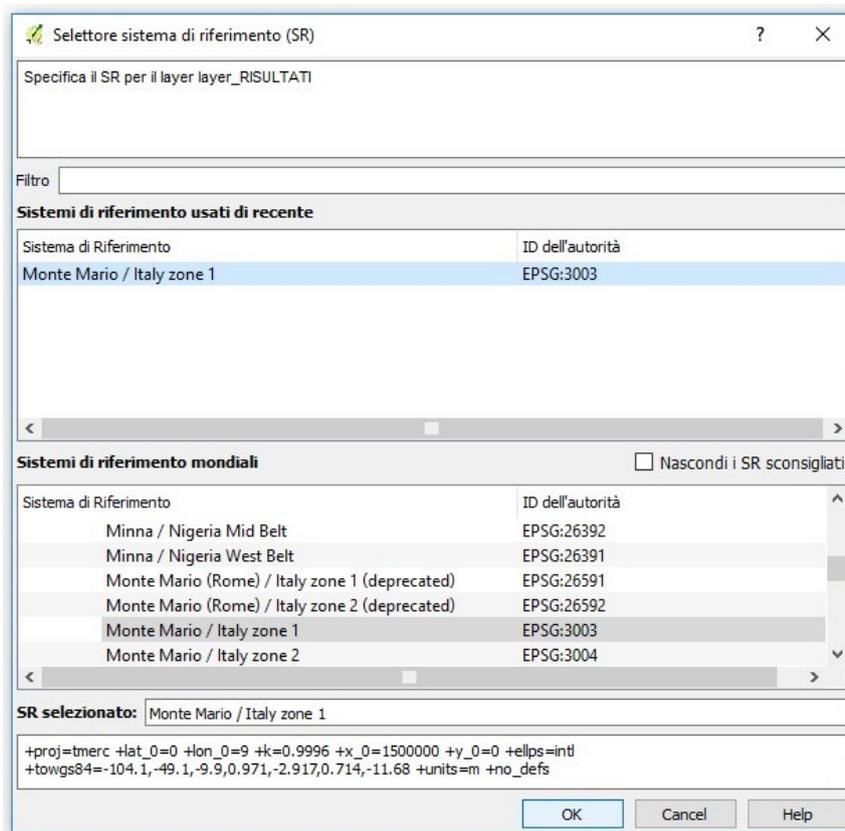


Figura 4.73: Fase: 3 - Importazione grafica risultati su QGIS

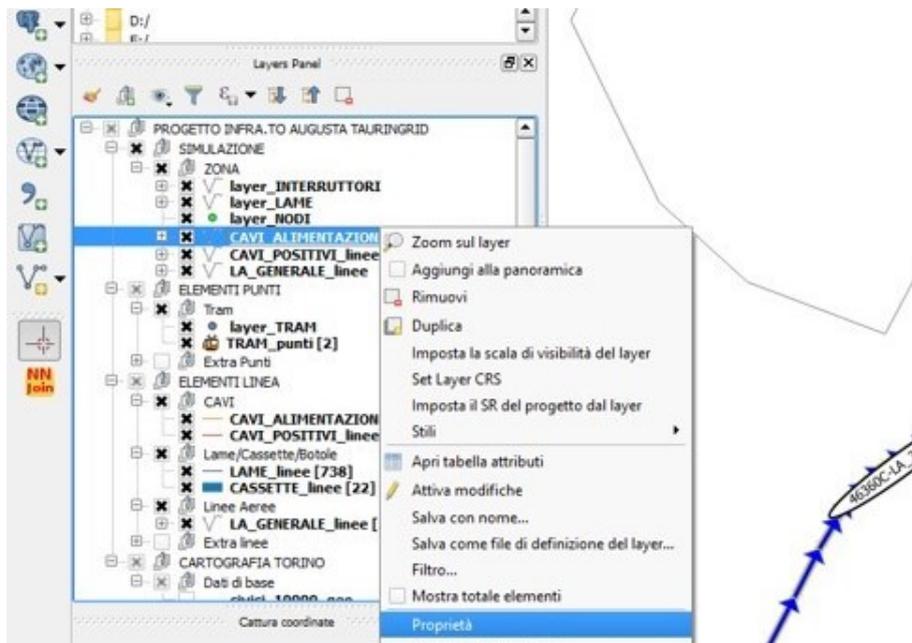


Figura 4.74: Fase: 1 - Gestione proprietà vettori su QGIS

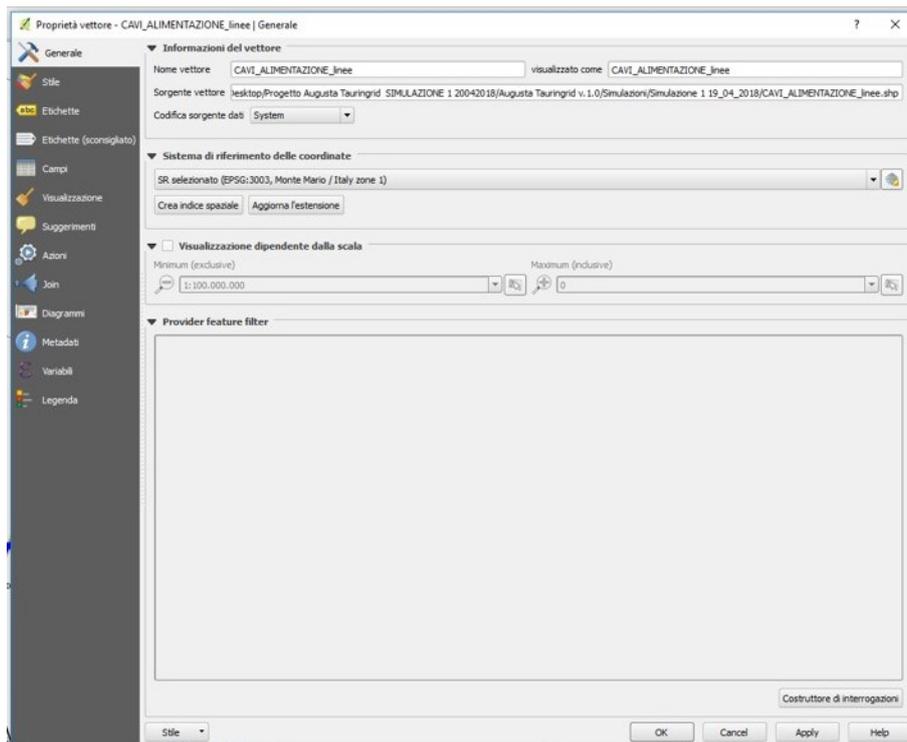


Figura 4.75: Fase: 2 - Gestione proprietà vettori su QGIS

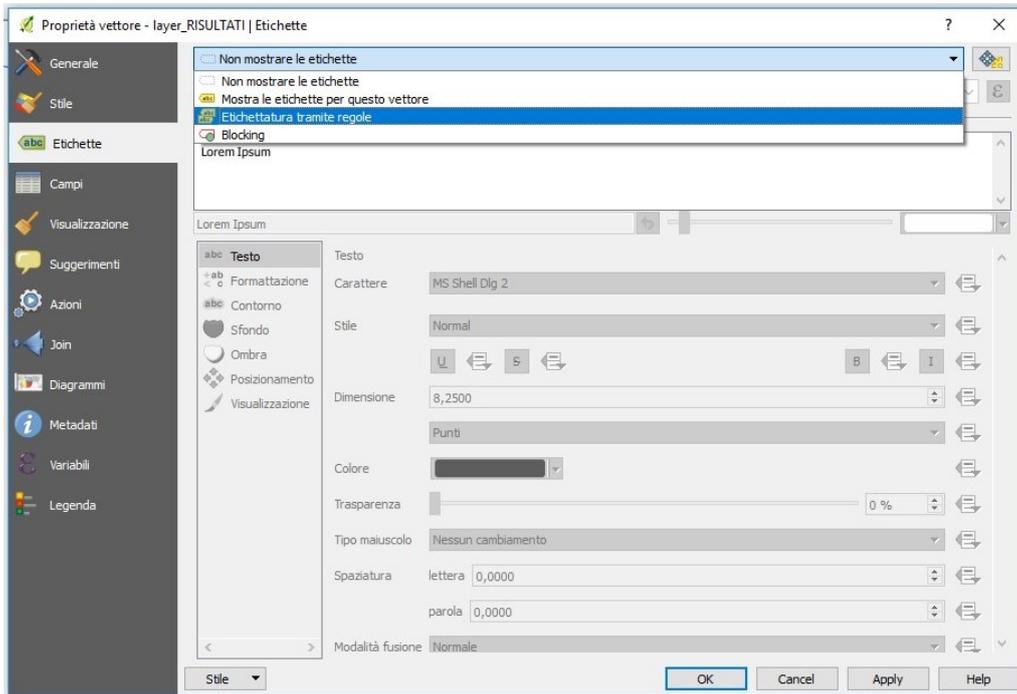


Figura 4.76: Fase: 3 - Gestione proprietà vettori su QGIS

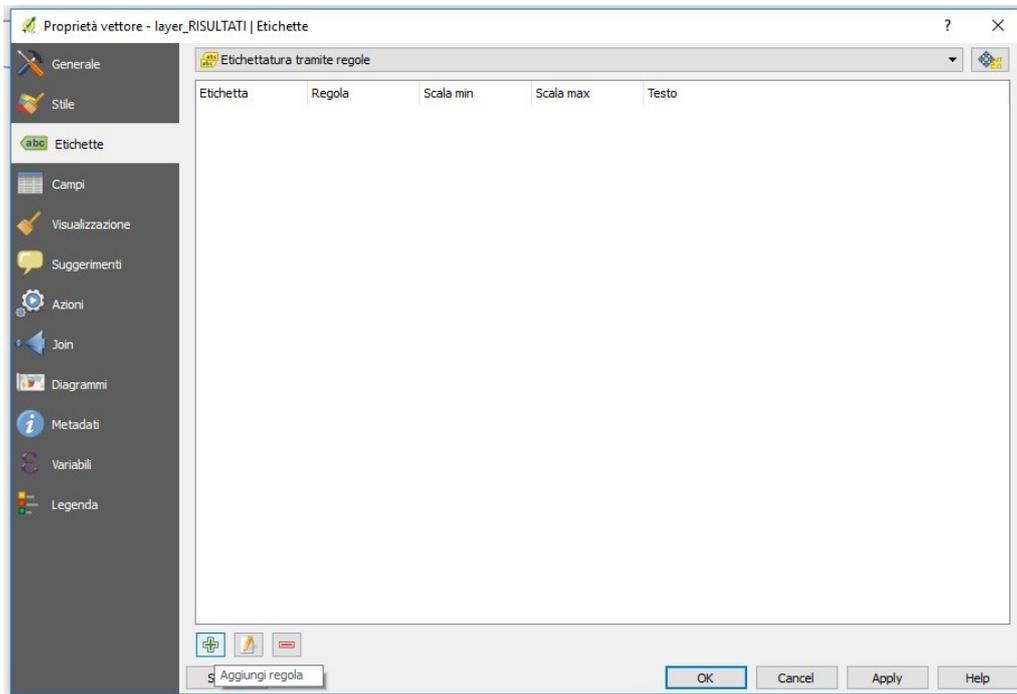


Figura 4.77: Fase: 4 - Gestione proprietà vettori su QGIS

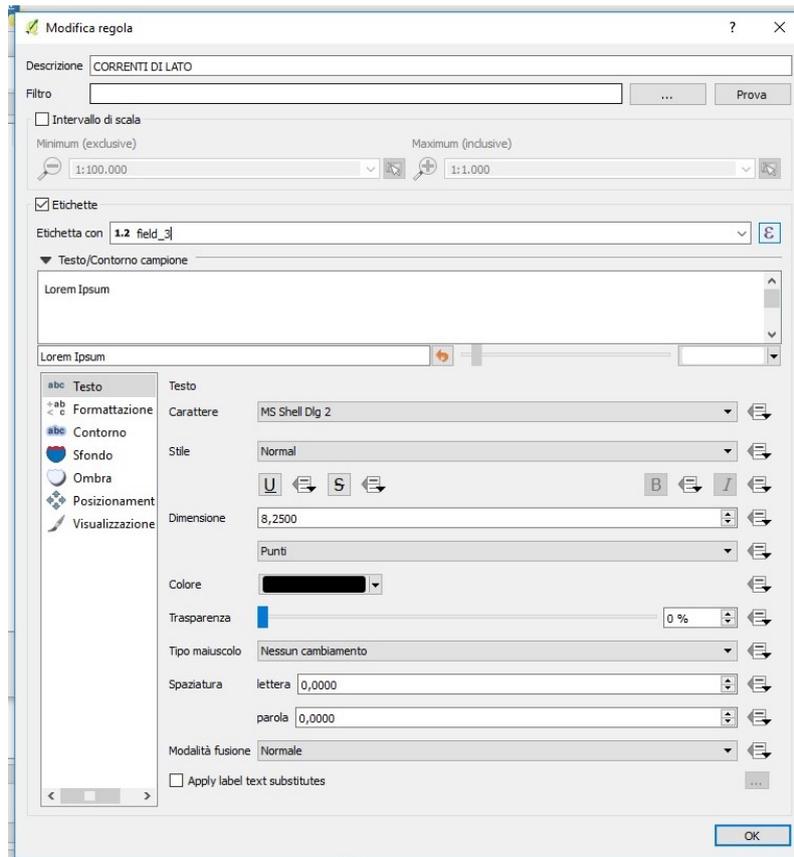


Figura 4.78: Fase: 5 - Gestione proprietà vettori su QGIS

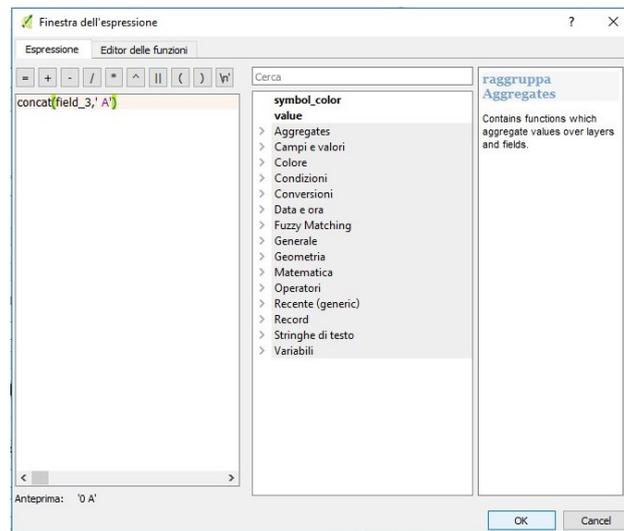


Figura 4.79: Fase: 6 - Gestione proprietà vettori su QGIS

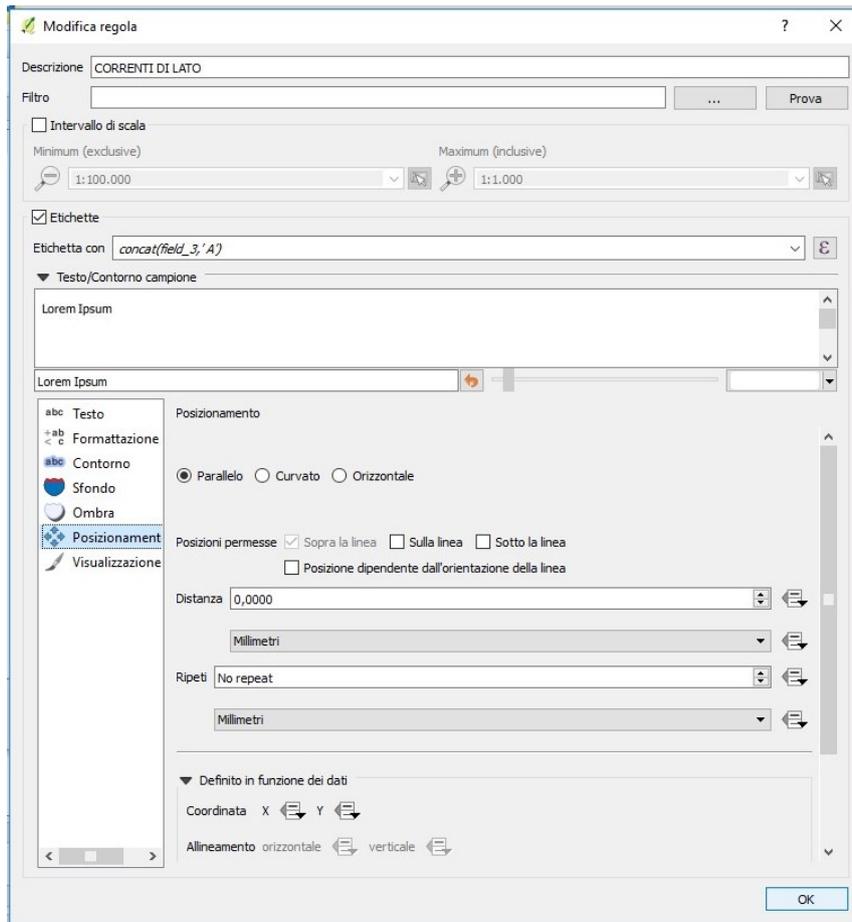


Figura 4.80: Fase: 7 - Gestione proprietà vettori su QGIS

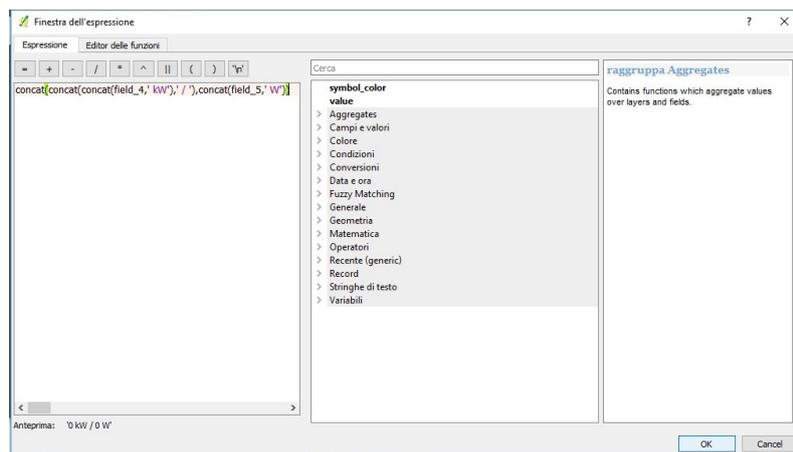


Figura 4.81: Fase: 8 - Gestione proprietà vettori su QGIS

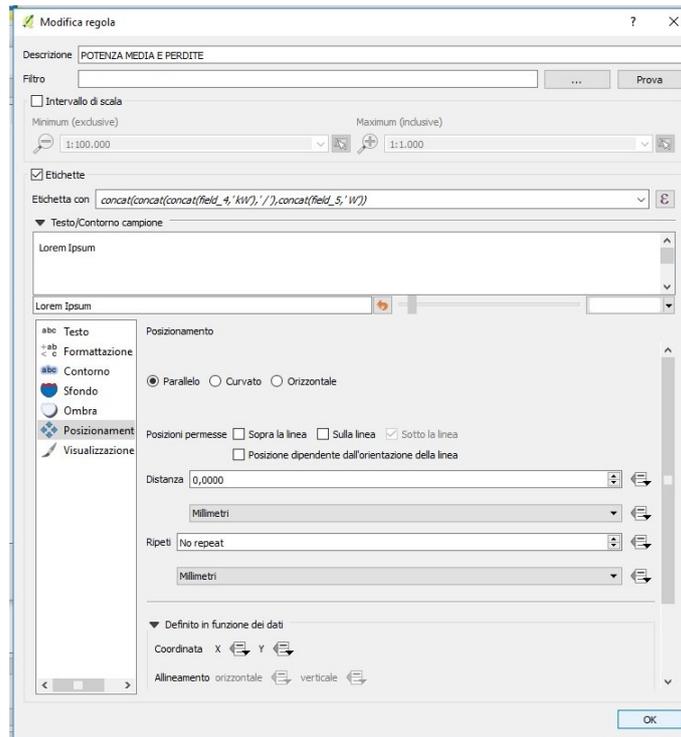


Figura 4.82: Fase: 9 - Gestione proprietà vettori su QGIS

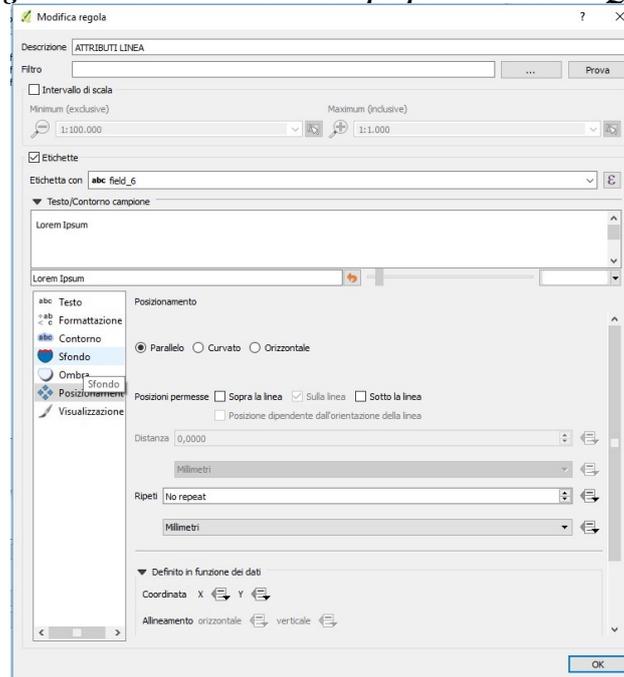


Figura 4.83: Fase: 10 - Gestione proprietà vettori su QGIS

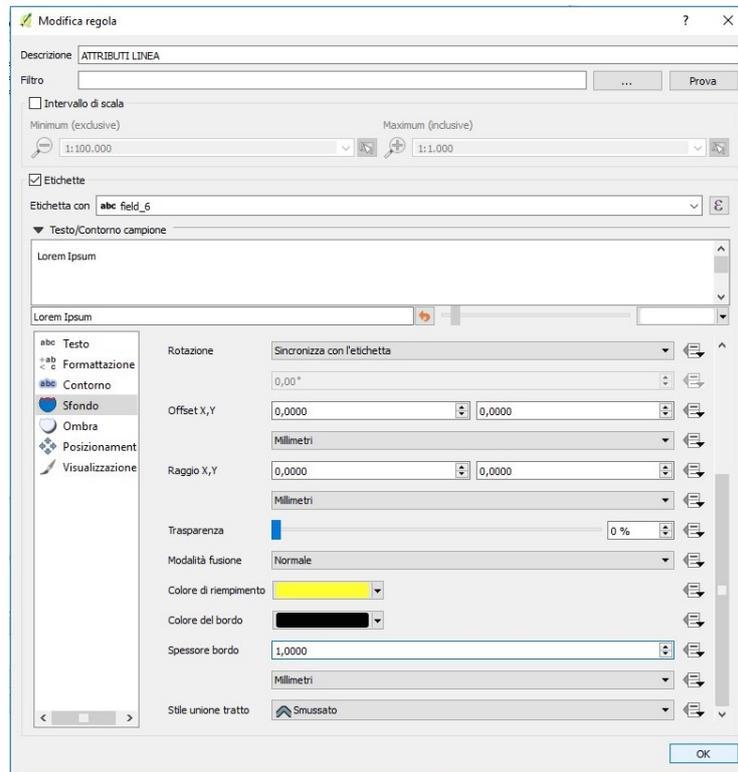


Figura 4.84: Fase: 11 - Gestione proprietà vettori su QGIS

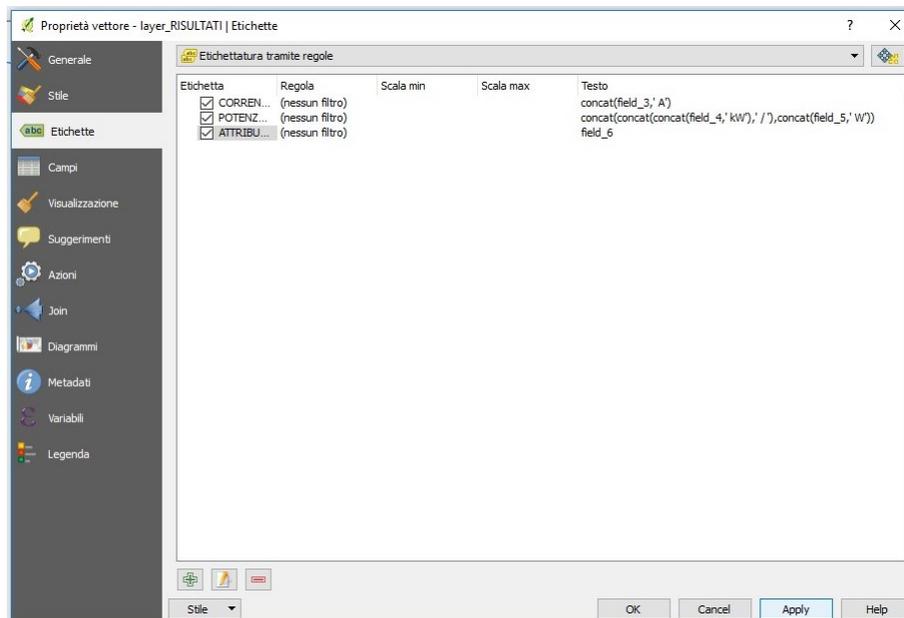


Figura 4.85: Fase: 12 - Gestione proprietà vettori su QGIS

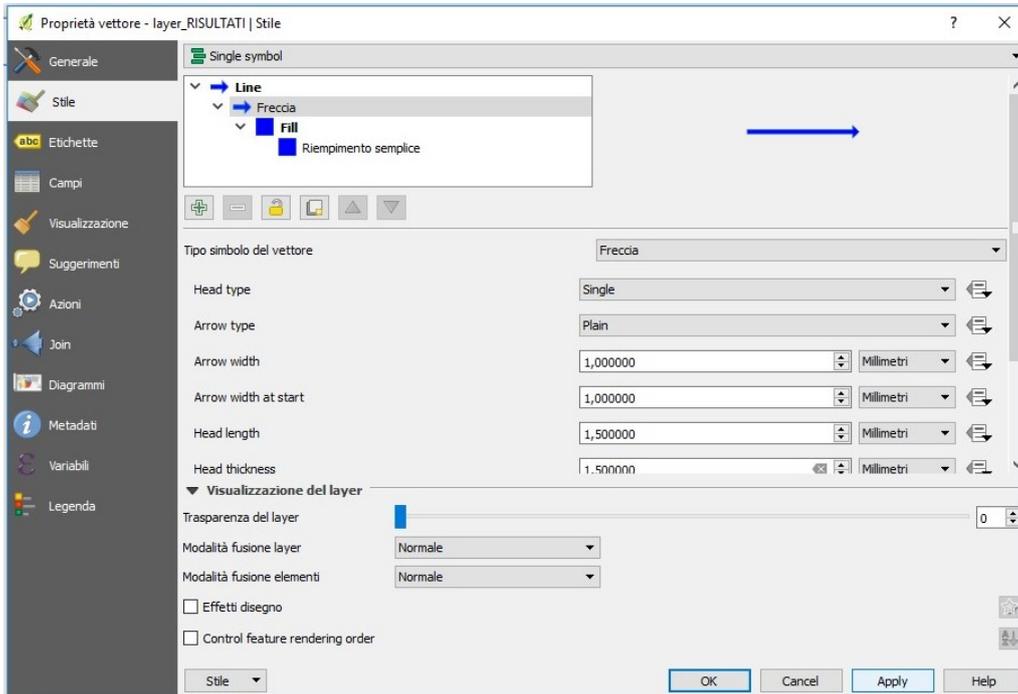


Figura 4.86: Fase: 13 - Gestione proprietà vettori su QGIS

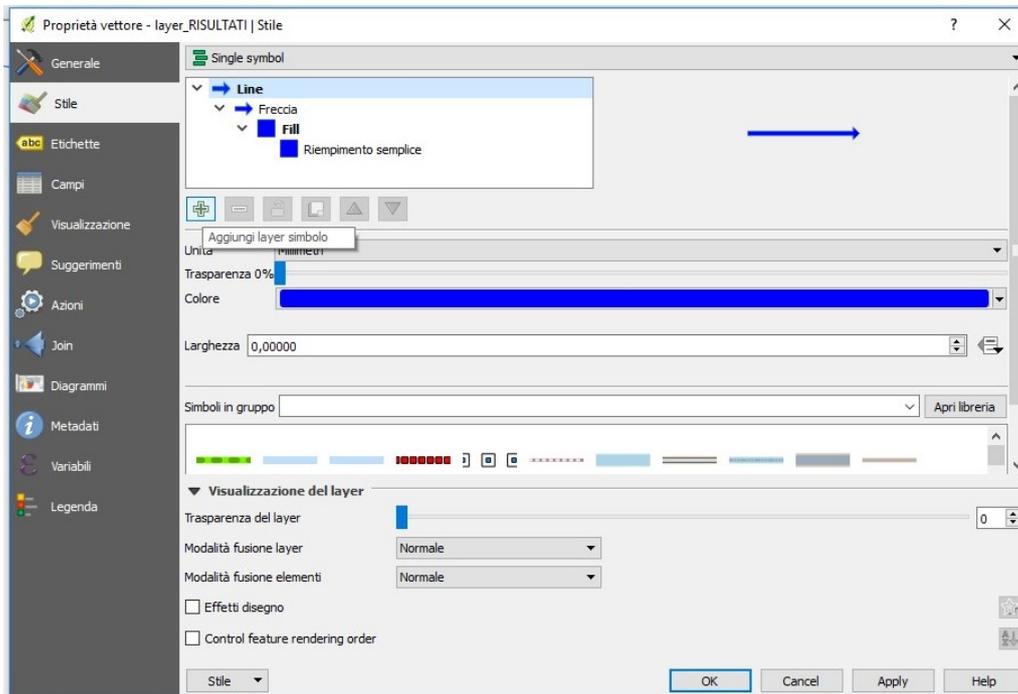


Figura 4.87: Fase: 14 - Gestione proprietà vettori su QGIS

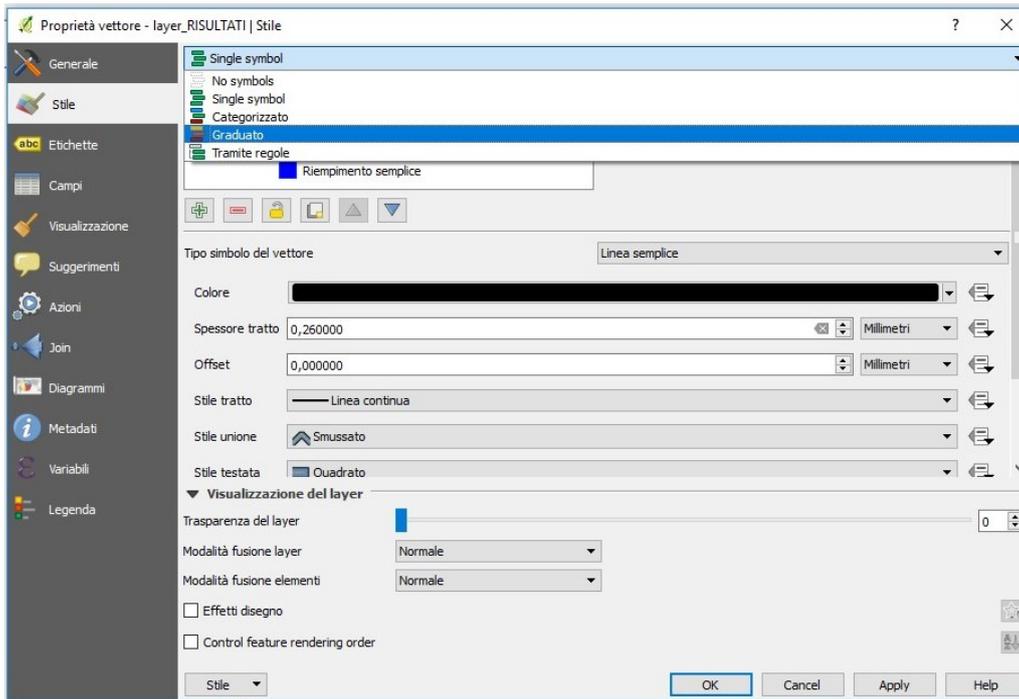


Figura 4.88: Fase: 15 - Gestione proprietà vettori su QGIS

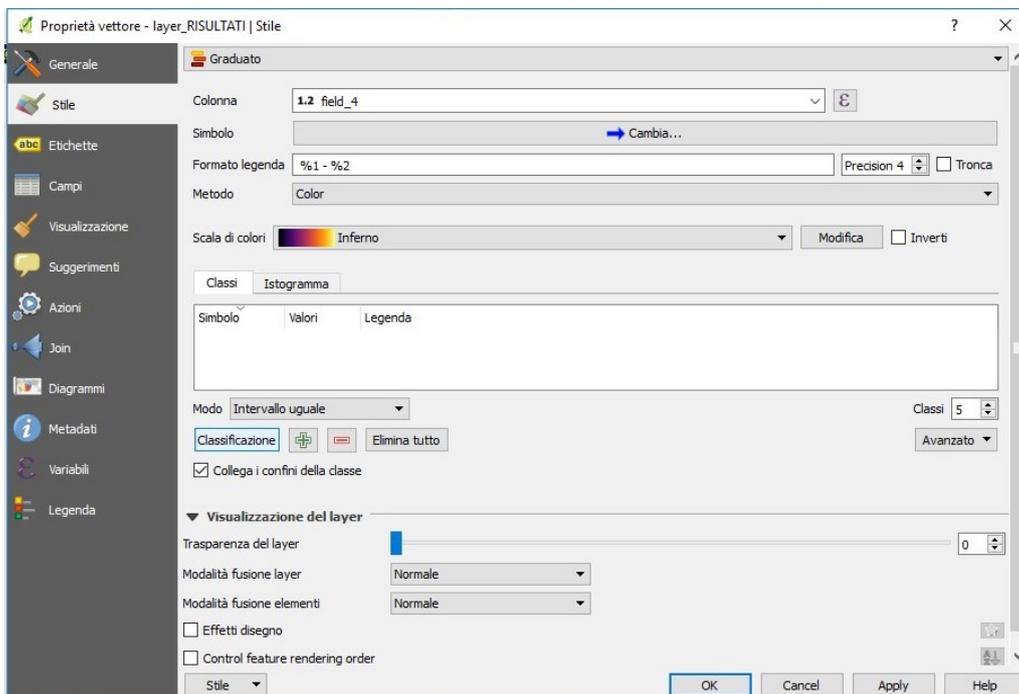


Figura 4.89: Fase: 16 - Gestione proprietà vettori su QGIS

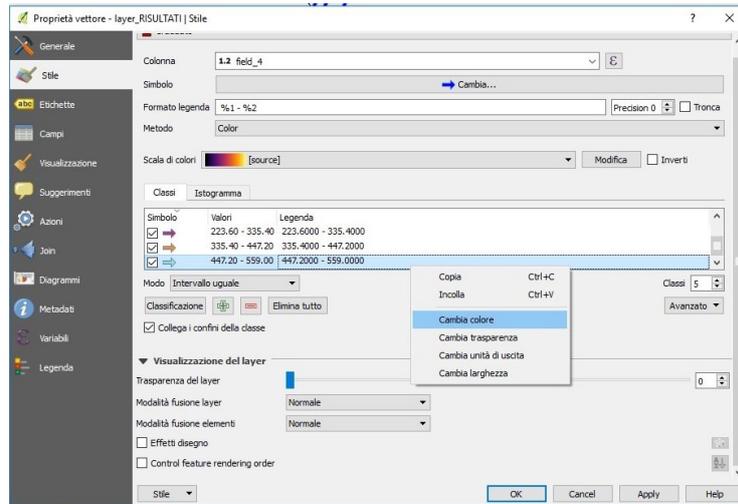


Figura 4.90: Fase: 17 - Gestione proprietà vettori su QGIS

2. Come visibile da figura 4.91 a figura 4.93 è possibile salvare uno stile predefinito per velocizzare il processo di importazione dei dati e se il nome del pacchetto di stile non varia o viene ridefinito in caso di variazioni, l'aggiornamento è automatizzato per ogni successiva modifica.

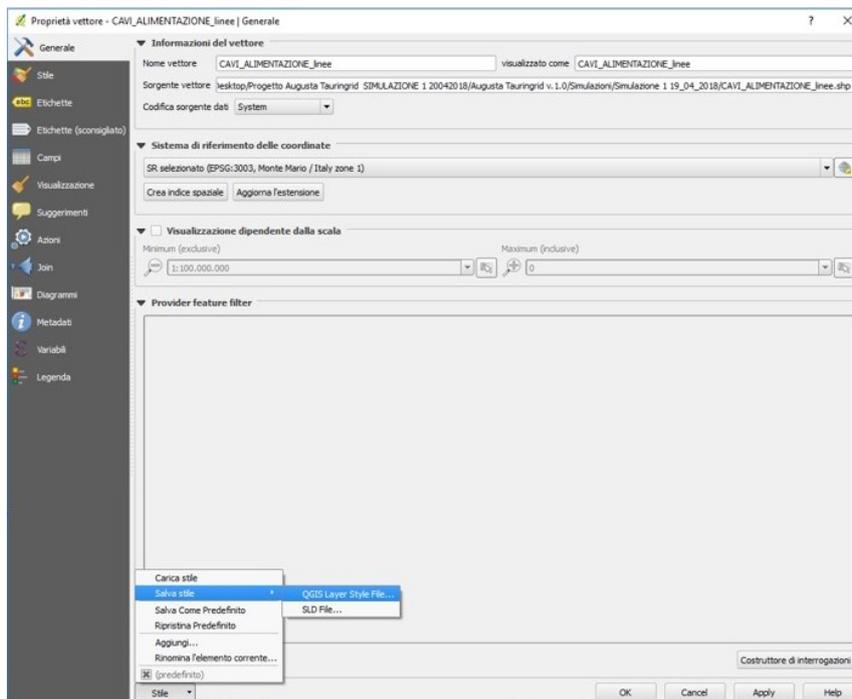


Figura 4.91: Fase: 1 - Stile pre-impostato grafica risultati su QGIS

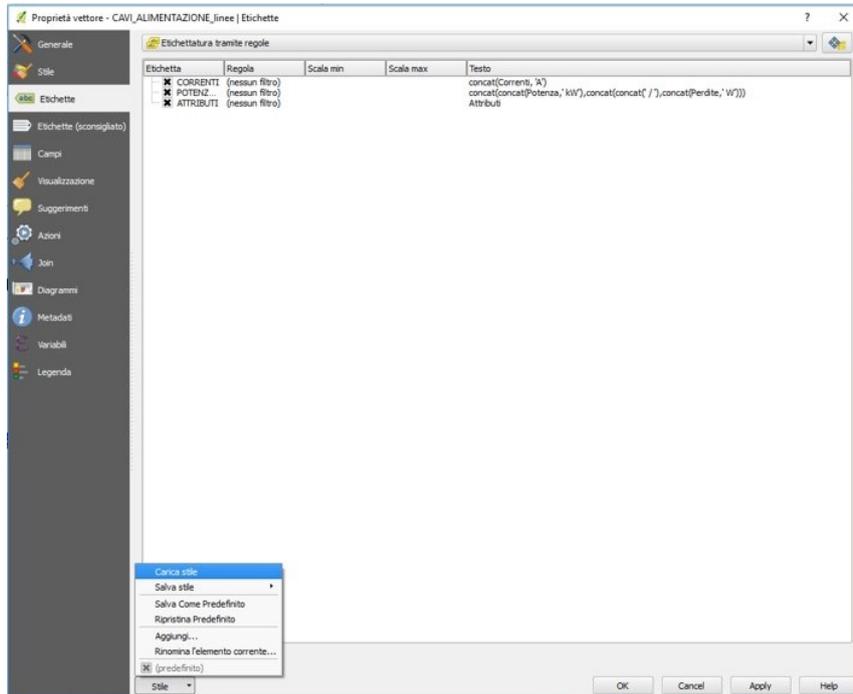


Figura 4.92: Fase: 2 - Stile pre-impostato grafica risultati su QGIS

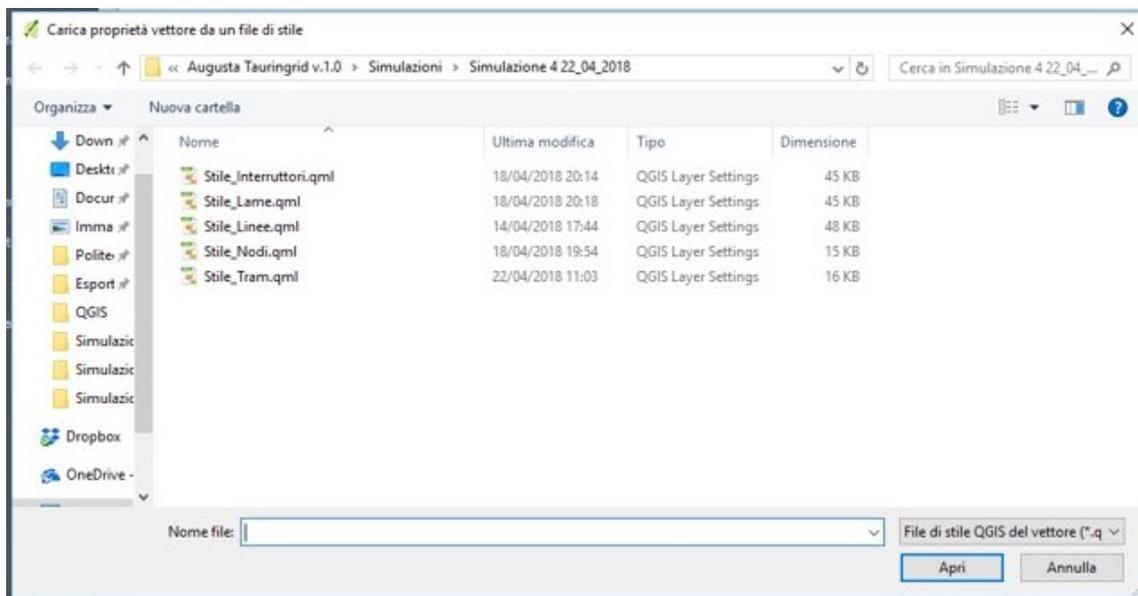


Figura 4.93: Fase: 3 - Stile pre-impostato grafica risultati su QGIS

Capitolo 5

Simulazione di ‘Augusta Tauringrid’

Si effettuano simulazioni in **differenti scenari** per analizzarne i risultati e confrontare il tutto nel capitolo delle conclusioni. Si considerano 4 scenari visibili in figura 5.1 di cui il primo in condizioni ordinarie di funzionamento della rete e gli ultimi 3 in condizioni di guasto della rete dovuto a cortocircuito di tipo franco [3]. Da applicazioni sperimentali in funzione delle condizioni al contorno su cui verte lo scenario le correnti assorbite del tram rientrano in un range prestabilito con valori di gradiente durante la fase di accelerazione tra 1 e 5 A/ms [8]. Si pone per la simulazione la corrente assorbita dalla motrice pari a 1000 A. Si modella la sottostazione elettrica con un modello equivalente Thevenin dove la resistenza equivalente serie raddrizzatore e trasformatore viene determinata tramite misurazione sperimentale e posta pari a 0.0167Ω [7].

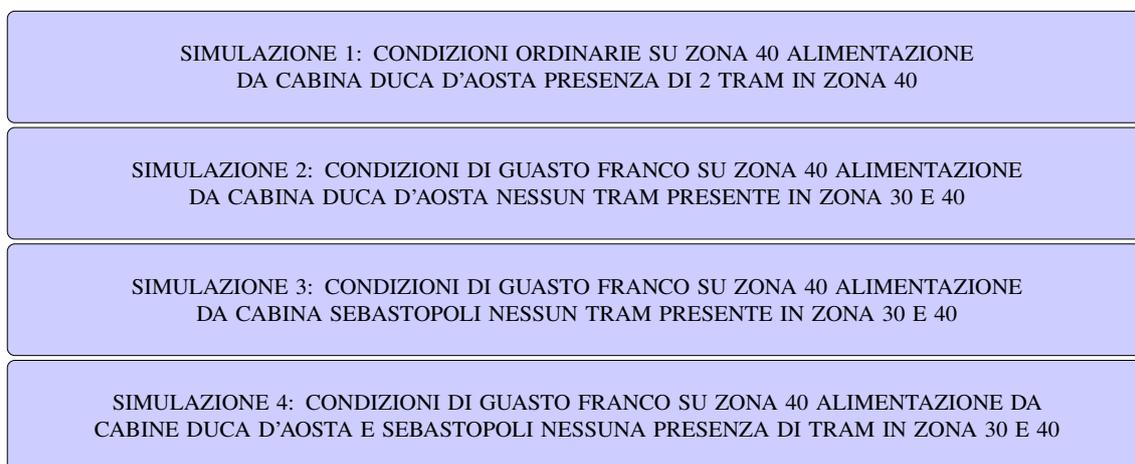


Figura 5.1: Scenari Simulati

Per valutare la veridicità dei risultati si è definito un indice d'errore pari alla somma delle correnti entranti ed uscenti in ciascun nodo.

Questo indice si rifà direttamente alla legge di Kirchhoff delle correnti che in un sistema matriciale si può scrivere come riferimento all'equazione 5.1. Maggiore è questo indice più elevati saranno gli errori del software Augusta Tauringrid. Per tutti gli scenari analizzati il valore dell'indice coefficiente d'errore *C.E.* è inferiore a 0.006.

$$A * j = 0 \tag{5.1}$$

Dai dati sperimentali emerge che tale condizione è pienamente rispettata per ciascuna delle simulazioni che verranno illustrate a seguire. Nelle simulazioni viene trascurato il valore della resistenza di binario che ridurrebbe ulteriormente i valori della corrente avvicinando la modellistica di un sistema elettrico ideale ad una realtà applicativa.

5.1 Scenario 1: Rete elettrica tranviaria zona 40 in condizioni ordinarie

5.1.1 Analisi dei Risultati

1. Si analizza la **zona 40** nelle vicinanze del Politecnico di Torino in cui viene esclusivamente attivata la cabina Duca d'Aosta e vengono posti 2 tram prima in accelerazione e successivamente in frenata. Lo scenario di simulazione è illustrato in figura 5.2. Come è possibile notare anche una singola zona risulta essere un sistema notevolmente complesso con miriadi di connessioni intrinseche tra barrature in rame, cavi positivi, di alimentazione e linee aeree. Si analizzano i risultati visibili in figura 5.3 e figura 5.4 ed inoltre nello specifico si può verificare che entrambi i tram assorbano la corrente regolare di 1000 A come da specifiche imposte dalle variabili globali del codice. Si può riscontrare che la tensione ai nodi 1425 e 3947 dei 2 tram rispettivamente è di 638 V e di 653 V in frenata rigenerativa con immissione in rete, e di 562 V e di 547 V in accelerazione con assorbimento di corrente. Attraverso le heatmap preimpostate è possibile identificare i cavi o le linee attraversate eccessivamente da corrente con il colore rosso piuttosto che blu.



(a) Panoramica QGIS



(b) Panoramica Satellite

Figura 5.2: Simulazione 1

5.2.2 Verifica dei risultati

- Analizzo per completezza il circuito semplificato che modella la struttura del sistema elettrico illustrato in figura 5.6 le cui proprietà elettriche sono riassunte in tabella 5.1.

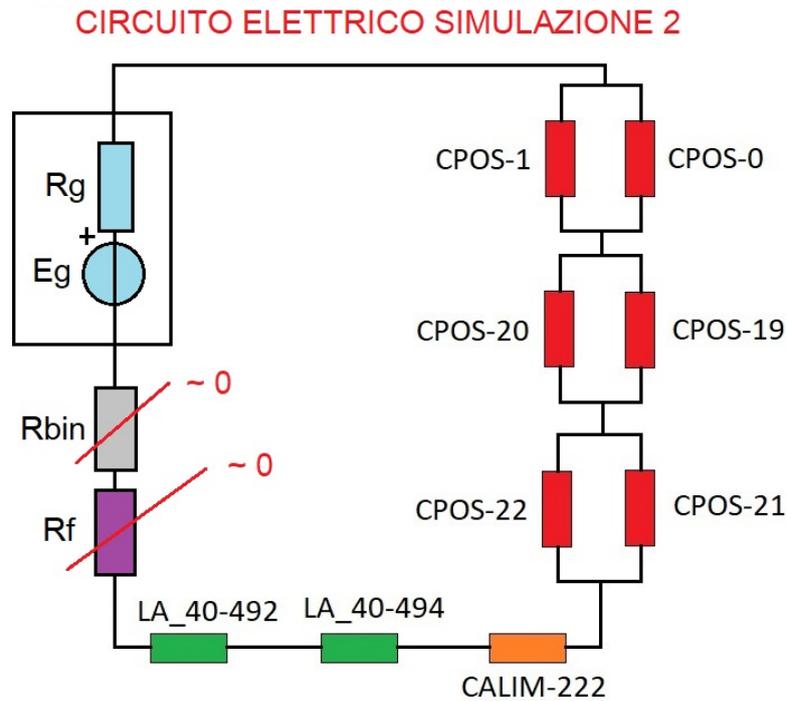


Figura 5.6: Analisi circuitale simulazione 2

ID	R[mΩ]	S[mmq]	L[m]
CPOS-0	4	1000	235
CPOS-1	4	1000	235
CPOS-19	10	1000	586
CPOS-20	27	1000	1570
CPOS-21	27	1000	1570
CPOS-22	2	1000	112
CALIM-222	0.17	500	5
LA_40-494	9.7	95	54
LA_40-492	1.1	95	6

Tabella 5.1: Tabella simulazione 2

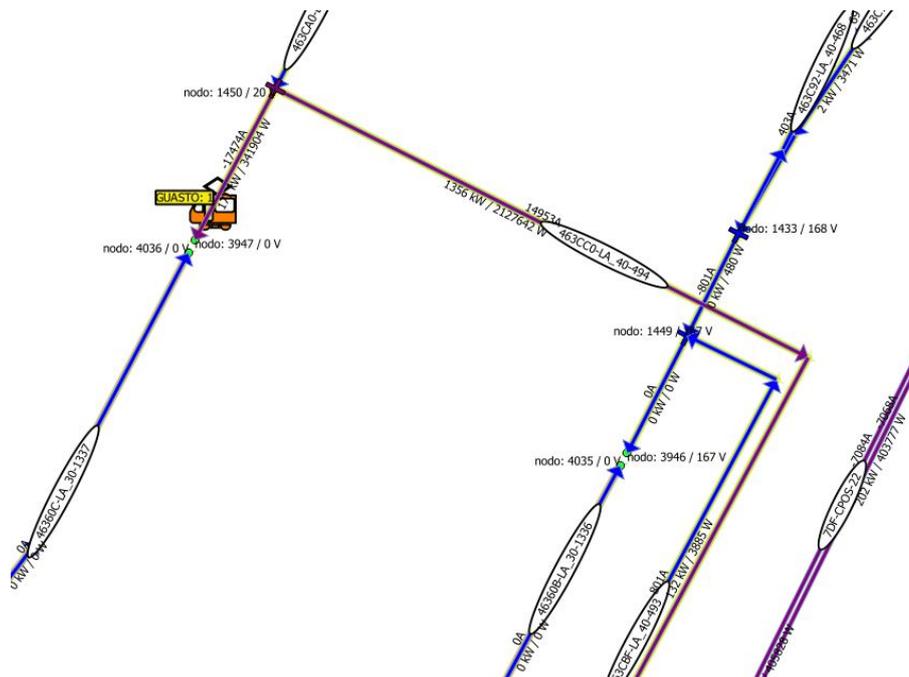


Figura 5.7: Simulazione 2: Visualizzazione dei risultati 1

3. Risoluzione analitica del circuito semplificato secondo l'equazione 5.2:

$$I_{cc}' = \frac{600V}{(16.7 + 2 + 7.3 + 1.9 + 0.17 + 9.7 + 1.1)m\Omega} \approx \frac{600V}{39m\Omega} \approx 15kA \quad (5.2)$$

4. Dal confronto con i valori del software si può constatare la validità ed affidabilità del software sul calcolo automatico delle correnti di cortocircuito della rete. Vi è una differenza sostanziale tra calcoli teorici su modello semplificato e calcolo automatico della corrente tramite software dato dal fatto che si trascura la complessità della rete.

5.3 Scenario 3: Rete elettrica tranviaria zona circostante 30 Politecnico di Torino in condizioni di guasto a terra

5.3.1 Analisi dei risultati

1. Si ipotizza un guasto franco nella zona 30 e nelle medesime condizioni esaminate precedentemente di alimentazione tranne per la presenza di motrici sulla zona che vengono escluse e sostituite da un ramo di cortocircuito e dalla cabina di alimentazione che passa da Duca d'Aosta a Sebastopoli come visibile in figura 5.8 e figura 5.9. La corrente di cortocircuito sul ramo interessato da guasto franco risulta essere di 15 kA come visibile in figura 5.11.



(a) Panoramica QGIS 1



(b) Panoramica QGIS 2

Figura 5.8: Simulazione 3



Figura 5.9: Simulazione 3: Panoramica Satellite

5.3.2 Verifica dei risultati

- Analizzo per completezza il circuito semplificato che modella il sistema illustrato in figura 5.10 e la tabella 5.2 che riassume le proprietà elettriche del circuito.

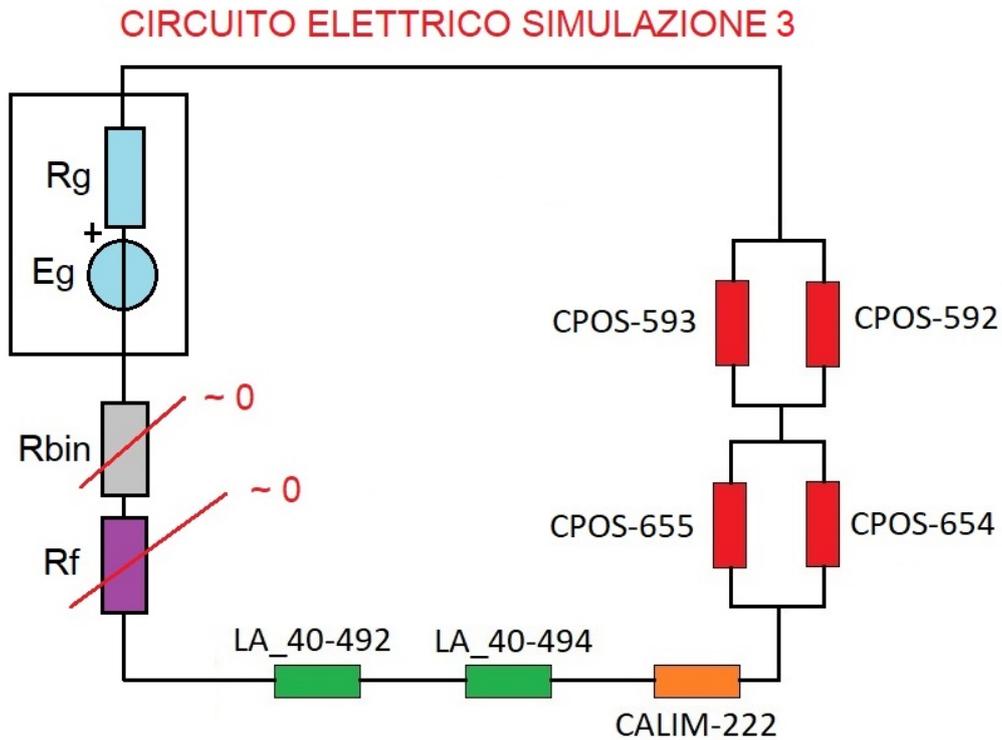


Figura 5.10: Analisi circuitale simulazione 3

ID	R[mΩ]	S[mmq]	L[m]
CPOS-592	6	1000	352
CPOS-593	13	500	382
CPOS-654	11	500	308
CPOS-655	14	500	411
CALIM-222	0.17	500	5
LA_40-494	9.7	95	54
LA_40-492	1.1	95	6

Tabella 5.2: Tabella simulazione 3

3. Risoluzione analitica del circuito secondo l'equazione 5.3:

$$I_{cc'} = \frac{600V}{(16.7 + 4.1 + 6 + 0.17 + 9.7 + 1.1)m\Omega} \approx \frac{600V}{38m\Omega} \approx 16kA \quad (5.3)$$

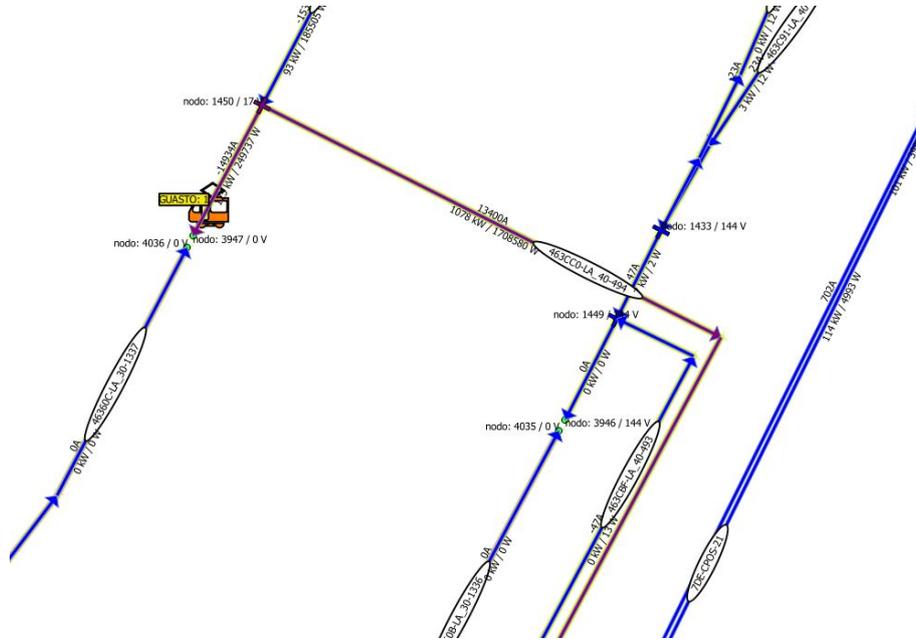


Figura 5.11: Simulazione 3: Visualizzazione dei risultati 1

4. Dal confronto con i valori del software si può constatare la validità ed affidabilità del software sul calcolo automatico delle correnti di cortocircuito della rete. Vi è una differenza sostanziale tra calcoli teorici su modello semplificato e calcolo automatico della corrente tramite software dato dal fatto che si trascura la complessità della rete.

5.4 Scenario 4: Rete elettrica tranviaria in condizioni di guasto a terra nella zona 40 alimentata da due sottostazioni elettriche

5.4.1 Analisi dei risultati

1. A partire dai risultati delle simulazioni precedenti mantenendo la posizione del guasto franco la medesima per tutte le simulazioni, a variare in questa analisi è la presenza di alimentazione da entrambe le cabine Duca d'Aosta e Sebastopoli come visibile in figura 5.12 e figura 5.13. La corrente di cortocircuito sul ramo di interesse caratterizzato da guasto franco risulta essere di 26 kA come illustrato in figura 5.15.



Figura 5.12: Simulazione 4: Panoramica QGIS parte 1



Figura 5.13: Simulazione 4: Panoramica QGIS parte 2

5.4.2 Verifica dei risultati

2. Analizzo per completezza il circuito semplificato che modella il sistema illustrato in figura 5.14 e la tabella 5.3 che ne riassume le proprietà elettriche.

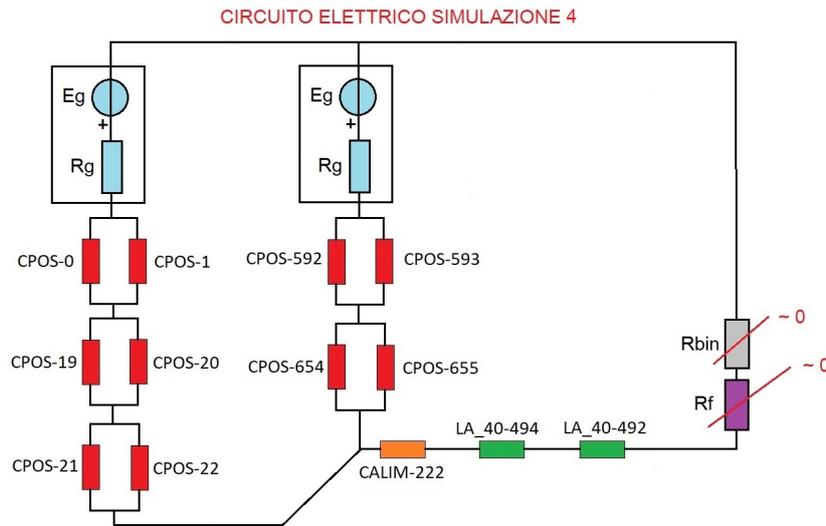


Figura 5.14: Analisi circuitale simulazione 4

ID	R[mΩ]	S[mmq]	L[m]
CPOS-0	4	1000	235
CPOS-1	4	1000	235
CPOS-19	10	1000	586
CPOS-20	27	1000	1570
CPOS-21	27	1000	1570
CPOS-22	2	1000	112
CPOS-592	6	1000	352
CPOS-593	13	500	382
CPOS-654	11	500	308
CPOS-655	14	500	411
CALIM-222	0.17	500	5
LA_40-494	9.7	95	54
LA_40-492	1.1	95	6

Tabella 5.3: Tabella simulazione 4

Capitolo 6

Conclusioni

E' stato sviluppato un software denominato Augusta Tauringrid secondo le specifiche del personale Infra.To, rispettando gli obiettivi imposti dalla tesi e simulando vari scenari applicativi per confrontare calcoli teorici con misure sperimentali. E' stato inoltre realizzato un modello della rete tranviaria di Torino in ambiente QGIS, che tramite interazione con codice matlab Augusta Tauringrid, mi consente di calcolare la corrente in ciascun lato e la tensione in ciascun nodo della rete e visualizzare in maniera intuitiva e semplificata i risultati. Il software è stato testato per quattro casi studio qui di seguito elencati:

1. Rete elettrica tranviaria zona 40 in condizioni ordinarie.
2. Rete elettrica tranviaria zona Politecnico di Torino in condizioni di guasto a terra.
3. Rete elettrica tranviaria zona circostante 30 Politecnico di Torino in condizioni di guasto a terra.
4. Rete elettrica tranviaria in condizioni di guasto a terra nella zona 40 alimentata da due sottostazioni elettriche.

Nel primo caso studio, si sono analizzate le specifiche di rete con l'ipotesi che simultaneamente due tram posti in due punti differenti della zona frenino o accelerino istantaneamente. Nei casi successivi si è scelto un punto della rete tale per cui si è applicato un guasto franco a terra e sono state analizzate le correnti di cortocircuito nella configurazione di alimentazione della rete da singola cabina di zona o da entrambe le cabine in prossimità del punto di guasto a terra. Per ciascuno dei casi studio è stata verificata la legge di Kirchhoff ai nodi. Nel primo caso di studio si è riscontrato che la tensione ai nodi

dei 2 tram rispettivamente è di 638 V e di 653 V in frenata rigenerativa con immissione in rete, e di 562 V e di 547 V in accelerazione con assorbimento di corrente. Dall'analisi dei risultati si ottiene che la differenza in modulo tra i risultati, rispettivamente in accelerazione e frenata, è di 15V. Il tram più distante dalla cabina ha tensione ai nodi in accelerazione inferiore a quello più prossimo, e superiore al medesimo in caso di frenata. Analizzando gli altri scenari si può constatare che la corrente di cortocircuito in condizioni di guasto franco a terra passando dall'alimentazione di una singola cabina a quella di duplice, ha un incremento percentuale nei rispettivi casi di alimentazione da zona più prossima e zona più lontana del 44% e del 73%. Inoltre risulta che nel caso di alimentazione da cabina più prossima al punto di guasto, la corrente di cortocircuito risulti più elevata rispetto al caso di alimentazione da cabina più distante dal medesimo punto. In passato, sono state effettuate delle misure sperimentali tali per cui, la corrente di cortocircuito è tipicamente nell'intervallo compreso tra 500 A e 14250 A. Tenuto conto che nel calcolo si è trascurata la resistenza di binario, i valori delle correnti di cortocircuito calcolati negli scenari in condizioni di guasto a terra sono compatibili con i valori misurati [6]. Il progetto di tesi rispetta le specifiche in termini di gestione e modifica dei dati caratterizzanti la rete ed offre una visualizzazione grafica dei risultati.

Bibliografia

- [1] URL: <http://www.tramditorino.it/>.
- [2] URL: <http://www.infrato.it/it/>.
- [3] E.Pons. “Analisi del guasto a terra nelle reti in media tensione, al fine della definizione di Impianto di Terra Globale”. In: *Tesi di Laurea* (2004), pp. 19–33.
- [4] *Fondamenti di teoria dei circuiti*. Charles A. Desoer Ernest S. Kuh, 2014.
- [5] F.Scolamiero. “Ottimizzazione dei sistemi di protezione della rete tranviaria torinese”. In: *Tesi di Laurea* (2016), pp. 5–25.
- [6] L.Bramardi. “Studio sperimentale delle correnti ordinarie e di guasto nella rete tranviaria torinese per l’ottimizzazione delle protezioni”. In: *Tesi di Laurea* (2017), pp. 184–214.
- [7] E.Pons R.Tommasini P.Colella. “Fault Current Detection and Dangerous Voltages in DC Urban Rail Traction Systems”. In: 53 (2017), pp. 4109–4115.
- [8] E.Pons P.Colella R.Rizzoli R.Tommasini. “Distinguishing short circuit and normal operation currents in DC urban light railway systems”. In: (2017).

Indice delle figure

2.1	Tipologie di motrici	3
3.1	Schema concettuale funzionamento software Simulatore di rete tranviaria	6
3.2	Schermata Software Applicazione Autocad	7
3.3	Schermata Software Catasto Rete di Alimentazione	9
3.4	Schermata Software Simulatore di Rete Tranviaria	9
3.5	File Agg_EL_cartografia Torino InfraTo	10
4.1	Schema concettuale funzionamento software Augusta Tauringrid	11
4.2	Step 1: Accensione visualizzazione di tutti i layer	12
4.3	Step 2: Sblocco di tutti i layer bloccati	13
4.4	Step 3: Scongelo di tutti i layer bloccati	13
4.5	Step 4: Fase 1 - Comando Percorri i layer	14
4.6	Step 4: Fase 2 - Comando Percorri i layer	15
4.7	Step 4: Fase 3 - Comando Percorri i layer	15
4.8	Step 4: Fase 4 - Comando Percorri i layer	16
4.9	Step 4: Fase 5 - Comando Blocca	16
4.10	Step 4: Fase 6 - Salvataggio CASSETTE.dwg	17
4.11	Step 5: Esportazione con estensione .dxf	17
4.12	Step 1: Installazione Plugin	18
4.13	Step 2: Installazione Plugin	19
4.14	Step 3: Installazione Plugin	19
4.15	Step 4: Installazione Plugin	20
4.16	Step 5: Installazione Plugin	20
4.17	Step 1: Configurazione Plugin	21
4.18	Step 2: Configurazione Plugin	21
4.19	Step 1: Installazione Plugin	22
4.20	Step 2: Installazione Plugin	22
4.21	Step 1: Aggiunta di un vettore layer	24
4.22	Step 2: Aggiunta di un vettore layer	25
4.23	Step 3: Aggiunta di un vettore layer	25
4.24	Step 4: Aggiunta di un vettore layer	26

4.25	Step 1: Gestione di un vettore layer	27
4.26	Step 2: Gestione di un vettore layer	27
4.27	Panoramica Augusta Tauringrid 1	28
4.28	Panoramica Augusta Tauringrid 2	28
4.29	Panoramica Augusta Tauringrid 3	29
4.30	Step: 1 - Plugin AutoFields	29
4.31	Step: 2 - Plugin AutoFields	30
4.32	Step: 3 - Plugin AutoFields	30
4.33	Step: 4 - Codici Plugin AutoFields	31
4.34	Step: 4 - Codici Plugin AutoFields	31
4.35	Step: 4 - Codici Plugin AutoFields	32
4.36	Step: 1 - Aggiunta elementi	32
4.37	Step: 2 - Aggiunta elementi	33
4.38	Step: 3 - Aggiunta elementi	33
4.39	Step: 4 - Aggiunta elementi	34
4.40	Step: 5 - Aggiunta elementi	34
4.41	Step: 6 - Aggiunta elementi	35
4.42	Step: 7 - Aggiunta elementi	35
4.43	Step: 8 - Aggiunta elementi	36
4.44	Step: 9 - Aggiunta elementi	36
4.45	Step: 10 - Aggiunta elementi	37
4.46	Misurazione entità	37
4.47	Step: 1 - Modifica elementi	38
4.48	Step: 2 - Modifica elementi	39
4.49	Step: 3 - Modifica elementi	39
4.50	Step: 4 - Modifica elementi	40
4.51	Step: 5 - Modifica elementi	40
4.52	Fase: 1 - Esportazione .CSV	41
4.53	Fase: 2 - Esportazione .CSV	42
4.54	Fase: 3 - Esportazione .CSV	42
4.55	Step: 1 - Interfaccia refresh	44
4.56	Step: 2 - Interfaccia refresh	44
4.57	Step: 3 - Interfaccia refresh	45
4.58	Step: 4 - Interfaccia refresh	45
4.59	Step: 5 - Interfaccia refresh	46
4.60	Step: 6 - Interfaccia refresh	46
4.61	Step: 7 - Interfaccia refresh	47
4.62	Step: 8 - Interfaccia refresh	47
4.63	Step: 9 - Interfaccia refresh	48
4.64	Step: 10 - Interfaccia refresh	48
4.65	Step: 11 - Interfaccia refresh	49

4.66	Step: 1 - Run Augusta Tauringrid	49
4.67	Step: 2 - Run Augusta Tauringrid	50
4.68	Step: 3 - Run Augusta Tauringrid	50
4.69	Logo Software Augusta Tauringrid Bertolone Citin Lorenzo	51
4.70	Schema a blocchi logica di funzionamento software Augusta Tauringrid	52
4.71	Fase: 1 - Importazione grafica risultati su QGIS	53
4.72	Fase: 2 - Importazione grafica risultati su QGIS	54
4.73	Fase: 3 - Importazione grafica risultati su QGIS	54
4.74	Fase: 1 - Gestione proprietà vettori su QGIS	55
4.75	Fase: 2 - Gestione proprietà vettori su QGIS	55
4.76	Fase: 3 - Gestione proprietà vettori su QGIS	56
4.77	Fase: 4 - Gestione proprietà vettori su QGIS	56
4.78	Fase: 5 - Gestione proprietà vettori su QGIS	57
4.79	Fase: 6 - Gestione proprietà vettori su QGIS	57
4.80	Fase: 7 - Gestione proprietà vettori su QGIS	58
4.81	Fase: 8 - Gestione proprietà vettori su QGIS	58
4.82	Fase: 9 - Gestione proprietà vettori su QGIS	59
4.83	Fase: 10 - Gestione proprietà vettori su QGIS	59
4.84	Fase: 11 - Gestione proprietà vettori su QGIS	60
4.85	Fase: 12 - Gestione proprietà vettori su QGIS	60
4.86	Fase: 13 - Gestione proprietà vettori su QGIS	61
4.87	Fase: 14 - Gestione proprietà vettori su QGIS	61
4.88	Fase: 15 - Gestione proprietà vettori su QGIS	62
4.89	Fase: 16 - Gestione proprietà vettori su QGIS	62
4.90	Fase: 17 - Gestione proprietà vettori su QGIS	63
4.91	Fase: 1 - Stile pre-impostato grafica risultati su QGIS	63
4.92	Fase: 2 - Stile pre-impostato grafica risultati su QGIS	64
4.93	Fase: 3 - Stile pre-impostato grafica risultati su QGIS	64
5.1	Scenari Simulati	65
5.2	Simulazione 1	67
5.3	Simulazione 1: Visualizzazione dei risultati 1	68
5.4	Simulazione 1: Visualizzazione dei risultati 2	68
5.5	Simulazione 2: Panoramica QGIS	69
5.6	Analisi circuitale simulazione 2	70
5.7	Simulazione 2: Visualizzazione dei risultati 1	71
5.8	Simulazione 3	72
5.9	Simulazione 3: Panoramica Satellite	72
5.10	Analisi circuitale simulazione 3	73
5.11	Simulazione 3: Visualizzazione dei risultati 1	74
5.12	Simulazione 4: Panoramica QGIS parte 1	75

5.13	Simulazione 4: Panoramica QGIS parte 2	75
5.14	Analisi circuitale simulazione 4	76
5.15	Simulazione 4: Visualizzazione dei risultati 1	77

Appendice A

Codice Matlab Augusta Tauringrid

Si illustra nello specifico la struttura completa del codice suddiviso in due sezioni. La prima sezione è quella **BASE** dove si analizza la struttura della rete mentre la seconda di **TEST** analizza la sua configurazione:

1. Variabili Globali di sistema modificabili dal committente in funzione delle esigenze di simulazione richieste.

```
1 clear; %Pulisce le variabili di sistema
2 close all; %Chiude tutte le finestre aperte
3 clc; %Pulisce la command window di MATLAB
4 format long; %Imposta visibilita' dei risultati piu'accurata
5
6 % VARIABILI DI SISTEMA MODIFICABILI:
7
8 % Variabili globali
9 print_new = 0; %Variabile Stampa tabella aggiornamento
10 epsilon = 0.01; %Errore di tolleranza aggancio nodi matrice
   incidenze
11 pho = 1.68 * 10^(-8); %Resistivita' del rame
12 tolleranza_cassetta = 12;%Tolleranza imposta di CASSETTE
   sperimentale pari a 10[m] iscrizione circonferenza su
   perimetro
13 R_INTERRUTTORE_ON = 0.000001;%Resistenza imposta per avere
   interruttore in stato chiuso di ON
14 R_INTERRUTTORE_OFF = 1000000; %Resistenza imposta per avere
   interruttore in stato aperto di OFF
15
16 % Variabili Layer_CASSETTE
```

```

17     V_CASSETTE = 600; %Definizione tensione Cassette imposta a
        600[V]
18     R_CASSETTE = 0.0167; %Resistenza Ausiliari Cassette
        associata di gruppo(trrafo+convertitore)[ohm]
19
20 % Variabili Layer_LA;
21     diametro_LA = 95; %Diametro cavo standard convenzionale se
        non specificato [mm^2]
22
23 % Variabili Layer_LAME
24     lunghezza_LAME = 0.2; %Lunghezza delle lame[m]
25     sezione_LAME = 95;
    
```

2. E' possibile visualizzare il codice di importazione dei dati relativo al layer_CASSETTE. Nella seguente parte di codice è stato aperto e letto il .csv contenente i dati esportati da QGIS per poi suddividere il contenuto delle stringhe nei relativi campi di interesse e successivamente inserirli in tabella. Ogni parte del codice e' stata opportunamente commentata:

```

1 % Codice BASE – Parte 1 – IMPORTAZIONE DEI DATI DA QGIS
2 % Se la rete rimane invariata e si decide di modificare
        esclusivamente
3 % connessioni di lame e posizioni di tram e' sufficiente avviare
4 % esclusivamente il file di TEST altrimenti run sul seguente
        script .m per
5 % la prima volta.
6 % Parte pre-salvata
7 % Per questioni di funzionalita' e velocita' di codice si
        salvera' un .mat
8 % contenente tutti i calcoli relativi al layer:
9 % 1)CASSETTE: (NOME CABINA,-CST-,ID)
10 % 2)CAVI ALIMENTAZIONE: (CODICE UNIVOCO AUTOCAD,-CALIM-,ID)
11 % 3)CAVI POSITIVI: (CODICE UNIVOCO AUTOCAD,-CPOS-,ID)
12 % 4)LINEA AEREA: (CODICE UNIVOCO AUTOCAD,-LA-,ID)
13 % Parte iterabile ogni singola volta
14 % E successivamente nel seguente ordine all'importazione di:
15 % 5)TRAM: (CODICE UNIVOCO AUTOCAD,TR-,ID)
16 % 6)LAME: (CODICE UNIVOCO AUTOCAD,-LM-,ID)
17 % Sulla base della classificazione di id effettuiamo la lettura
        dei file.csv
    
```

```

18 % esportati da QGIS e da importare in MATLAB attraverso il
    seguente codice:
19 % 1)Apertura del layer_CASSETTE e salvataggio
20 % Vettore: Colonne_layer_CASSETTE
21 % Colonna 1: LAYER
22 % Colonna 2: ATTRIBUTI
23 % Colonna 3: CODICE UNIVOCO AUTOCAD
24 % Colonna 4: ORIGINE UNIVOCA AUTOCAD
25 % Colonna 5: ID
26 % Colonna 6: NOME CABINA
27 % Colonna 7: CODICE CABINA
28 % Colonna 8: X
29 % Colonna 9: Y
30 % Lettura del file .csv ed importazione in tabella
    Colonne_layer_CASSETTE
31 % Apertura del File .csv
32     fid = fopen('layer_CASSETTE.csv');
33 % Lettura e scansione in tabella
34     Colonne_layer_CASSETTE = textscan(fid, '%s%s%s%s%s%s%s%s',
        'delimiter', '\n');
35 % Chiusura del File .csv
36     fclose(fid);
37 % Aggiornamento dei campi e pulizia delle stringhe tramite
    apposito ciclo
38 % for delle Colonne_layer_CASSETTE
39 % La pulizia viene effettuata per estrapolare i dati necessari
    ed
40 % alleggerire il peso dei dati contenuti nelle tabelle
    attraverso apposite
41 % operazioni di sostituzione e delimitazione di stringa
42 % Inizializzazione variabile ausiliaria lettura dato che ogni
    colonna contiene un numero indefinito di righe
43     t = 0;
44 %Inizializzazione indice ausiliario lettura dati
45     k = 0;
46     for s=1:length(Colonne_layer_CASSETTE(1,:))
47         % Alla fine della lettura della serie di righe in una
            colonna riprendi da capo la colonna successiva
48         t = 0;
49         while(t<length(Colonne_layer_CASSETTE{: ,s}))
50

```

```
51         t = t+1;
52         k = k+1;
53
54         Lista_layer_CASSETTE{k,1} = Colonne_layer_CASSETTE{s
55             }{t};
56         % Strrep permette di sostituire parti di stringa con
57         % altre
58         Lista_layer_CASSETTE{k} = strrep(
59             Lista_layer_CASSETTE{k},',',' ');
60         % Regexp permette di differenziare tra loro elementi
61         % sulla base di un'altra stringa
62         Lista_layer_CASSETTE{k} = regexp(
63             Lista_layer_CASSETTE{k},' ','split');
64     end
65 end
66 % Variabile ausiliaria di verifica dove -1 giustifica la prima
67 % riga di intestazione
68 Num_Cassette = k-1;
69 % Sulla base della tabella Colonne_layer_CASSETTE trasferisco
70 % tutti i
71 % parametri di interesse in tab_layer_CASSETTE tramite
72 % iterazione di
73 % apposito ciclo for el'aiuto di una lista di elementi
74 % Variabili di inizializzazione del ciclo for
75 % Variabile ausiliaria che conteggia il numero di riga della
76 % tabella
77 k = 2;
78 % Variabile ausiliaria di tabella utilizzata come indice
79 % puntatore
80 w = 2;
81 % Inizializzazione ed intestazione della tabella
82 tab_layer_CASSETTE{1,1} = 'x';
83 tab_layer_CASSETTE{1,2} = 'y';
84 tab_layer_CASSETTE{1,6} = 'id';
85 % Salta l'intestazione della lista
86 for k = 2 : Num_Cassette+1
87     % Se rileva celle vuote in tabella fuoriesce dal ciclo
88     if(isempty(Lista_layer_CASSETTE{k,1}) == 1)
89         break; %Fuoriesce dal ciclo for
```

```

82     end
83     % Concateno numero identificativo univoco e id della
      cassetta
84     % Strcat permette di concatenare tra loro stringhe
85     tab_layer_CASSETTE{w,6} = strcat(Lista_layer_CASSETTE{k
      ,1}{1,6}, '-CST-');
86     tab_layer_CASSETTE{w,6} = strcat(tab_layer_CASSETTE{w
      ,6},Lista_layer_CASSETTE{k,1}{1,5});
87     % Str2double mi consente di tradurre il formato stringa
      in formato
88     % double
89     % Coordinata centrale cassetta x
90     tab_layer_CASSETTE{w,1} = str2double(
      Lista_layer_CASSETTE{k,1}{1,8});
91     % Coordinata centrale cassetta y
92     tab_layer_CASSETTE{w,2} = str2double(
      Lista_layer_CASSETTE{k,1}{1,9});
93     % Una volta eseguita l'operazione incremento la
      variabile
94     % ausiliaria di tabella puntatore
95     w = w+1;
96     end

```

3. Analogo ragionamento viene applicato al layer_CAVI_ALIMENTAZIONE dove in aggiunta viene sfruttato lo shapefile *.shp* per una più flessibile ed accurata importazione delle coordinate di geometria:

```

1  % 2)Apertura del layer_CAVI_ALIMENTAZIONE e salvataggio
2  % Vettore: Colonne_layer_CAVI_ALIMENTAZIONE
3  % Colonna 1: LAYER
4  % Colonna 2: ATTRIBUTI
5  % Colonna 3: SEZIONE
6  % Colonna 4: CODICE UNIVOCO AUTOCAD
7  % Colonna 5: ORIGINE UNIVOCA AUTOCAD
8  % Colonna 6: LUNGHEZZA [m]
9  % Colonna 7: INIZIO
10 % Colonna 8: FINE
11 % Colonna 9: ID
12 % Lettura del file .csv ed importazione in tabella
      Colonne_layer_CAVI_ALIMENTAZIONE
13 % Apertura del File .csv

```

```

14     fid = fopen('layer_CAVI_ALIMENTAZIONE.csv');
15 % Lettura e scansione in tabella
16     Colonne_layer_CAVI_ALIMENTAZIONE = textscan(fid, '%s%s%s%s%s
        %s%s%s%s%s%s%s%s', 'delimiter', ',');
17 % Chiusura del File .csv
18     fclose(fid);
19 % Lettura nome univoco e ID da tabella tramite iterazione di
        ciclo for
20 % apposito
21     name_CAVI_ALIMENTAZIONE = strcat(
        Colonne_layer_CAVI_ALIMENTAZIONE{4}, '-CALIM-');
22     name_CAVI_ALIMENTAZIONE = strcat(name_CAVI_ALIMENTAZIONE,
        Colonne_layer_CAVI_ALIMENTAZIONE{9});
23 % Definizione dell'intestazione ed ID
24     for j=2:length(name_CAVI_ALIMENTAZIONE)
25         tab_layer_CAVI_ALIMENTAZIONE{j,6} =
            name_CAVI_ALIMENTAZIONE{j,1};
26     end
27     tab_layer_CAVI_ALIMENTAZIONE{1,6} = 'id';
28 % Calcolo della Resistenza
29 % Lettura Lunghezza
30     lunghezza_CAVI_ALIMENTAZIONE = str2double(
        Colonne_layer_CAVI_ALIMENTAZIONE{6});
31 % Lettura Sezione
32     sezione_CAVI_ALIMENTAZIONE = str2double(
        Colonne_layer_CAVI_ALIMENTAZIONE{3});
33 % Espressa in [ohm]
34     R_CAVI_ALIMENTAZIONE = rdivide(lunghezza_CAVI_ALIMENTAZIONE,
        sezione_CAVI_ALIMENTAZIONE)*10^6 * pho;
35 % Definizione ed intestazione della Resistenza
36     for j=1:length(R_CAVI_ALIMENTAZIONE)
37         tab_layer_CAVI_ALIMENTAZIONE{j,5} = R_CAVI_ALIMENTAZIONE
            (j,1);
38     end
39     tab_layer_CAVI_ALIMENTAZIONE{1,5} = 'R';
40 % Lettura di Coordinate di Inizio Linea
41     [S_CALIM,A_CALIM] = shaperead('CAVI_ALIMENTAZIONE_linee.shp'
        );
42 % Definizione delle coordinate di Inizio Linea
43     for j = 1:length(S_CALIM)
44         x_start_CAVI_ALIMENTAZIONE = S_CALIM(j).X(1);

```

```

45     y_start_CAVI_ALIMENTAZIONE = S_CALIM(j).Y(1);
46     tab_layer_CAVI_ALIMENTAZIONE{j+1,1} =
        x_start_CAVI_ALIMENTAZIONE;
47     tab_layer_CAVI_ALIMENTAZIONE{j+1,2} =
        y_start_CAVI_ALIMENTAZIONE;
48     x_start_CAVI_ALIMENTAZIONE = 0;
49     y_start_CAVI_ALIMENTAZIONE = 0;
50     end
51 % Lettura di Coordinate di Fine Linea
52 % Definizione delle coordinate di Fine Linea
53     for j = 1:length(S_CALIM)
54         puntatore_end_CALIM = length(S_CALIM(j).X);
55         x_end_CAVI_ALIMENTAZIONE = S_CALIM(j).X(
            puntatore_end_CALIM-1);
56         y_end_CAVI_ALIMENTAZIONE = S_CALIM(j).Y(
            puntatore_end_CALIM-1);
57         tab_layer_CAVI_ALIMENTAZIONE{j+1,3} =
            x_end_CAVI_ALIMENTAZIONE;
58         tab_layer_CAVI_ALIMENTAZIONE{j+1,4} =
            y_end_CAVI_ALIMENTAZIONE;
59         x_end_CAVI_ALIMENTAZIONE = 0;
60         y_end_CAVI_ALIMENTAZIONE = 0;
61     end
62 % Intestazione delle coordinate una volta definite
63     tab_layer_CAVI_ALIMENTAZIONE{1,1} = 'x_start';
64     tab_layer_CAVI_ALIMENTAZIONE{1,2} = 'y_start';
65     tab_layer_CAVI_ALIMENTAZIONE{1,3} = 'x_end';
66     tab_layer_CAVI_ALIMENTAZIONE{1,4} = 'y_end';

```

4. Si commenta successivamente la parte di codice di importazione relativa al layer_CAVI_POSITIVI analoga al ragionamento precedente:

```

1 % 3)Apertura del layer_CAVI_POSITIVI e salvataggio
2 % Vettore: Colonne_layer_CAVI_POSITIVI
3 % Colonna 1: LAYER
4 % Colonna 2: ATTRIBUTI
5 % Colonna 3: SEZIONE
6 % Colonna 4: CODICE UNIVOCO AUTOCAD
7 % Colonna 5: ORIGINE UNIVOCA AUTOCAD
8 % Colonna 6: LUNGHEZZA [m]
9 % Colonna 7: INIZIO

```

```
10 % Colonna 8: FINE
11 % Colonna 9: ID
12 % Lettura del file .csv ed importazione in tabella
    Colonne_layer_CAVI_POSITIVI
13 % Apertura del File .csv
14     fid = fopen('layer_CAVI_POSITIVI.csv');
15 % Lettura e scansione in tabella
16     Colonne_layer_CAVI_POSITIVI = textscan(fid, '%s%s%s%s%s%s%s%
        s%s%s%s%s%s', 'delimiter', ',', ');
17 % Chiusura del File .csv
18     fclose(fid);
19 % Lettura nome univoco e ID da tabella tramite iterazione di
    ciclo for
20 % apposito
21     name_CAVI_POSITIVI = strcat(Colonne_layer_CAVI_POSITIVI{4}, '
        -CPOS-');
22     name_CAVI_POSITIVI = strcat(name_CAVI_POSITIVI,
        Colonne_layer_CAVI_POSITIVI{9});
23 % Definizione dell'intestazione ed ID
24     for j=2:length(name_CAVI_POSITIVI)
25         tab_layer_CAVI_POSITIVI{j,6} = name_CAVI_POSITIVI{j,1};
26     end
27     tab_layer_CAVI_POSITIVI{1,6} = 'id';
28 % Calcolo della Resistenza
29 % Lettura Lunghezza
30     lunghezza_CAVI_POSITIVI = str2double(
        Colonne_layer_CAVI_POSITIVI{6});
31 % Lettura Sezione
32     sezione_CAVI_POSITIVI = str2double(
        Colonne_layer_CAVI_POSITIVI{3});
33 % Espressa in [ohm]
34     R_CAVI_POSITIVI = rdivide(lunghezza_CAVI_POSITIVI,
        sezione_CAVI_POSITIVI)* 10^6 * pho;
35 % Definizione ed intestazione della Resistenza
36     for j=1:length(R_CAVI_POSITIVI)
37         tab_layer_CAVI_POSITIVI{j,5} = R_CAVI_POSITIVI(j,1);
38     end
39     tab_layer_CAVI_POSITIVI{1,5} = 'R';
40 % Lettura di Coordinate di Inizio Linea
41     [S_CPOS,A_CPOS] = shaperead('CAVI_POSITIVI_linee.shp');
42 % Definizione delle coordinate di Inizio Linea
```

```

43     for j = 1:length(S_CPOS)
44         x_start_CAVI_POSITIVI = S_CPOS(j).X(1);
45         y_start_CAVI_POSITIVI = S_CPOS(j).Y(1);
46         tab_layer_CAVI_POSITIVI{j+1,1} = x_start_CAVI_POSITIVI;
47         tab_layer_CAVI_POSITIVI{j+1,2} = y_start_CAVI_POSITIVI;
48         x_start_CAVI_POSITIVI = 0;
49         y_start_CAVI_POSITIVI = 0;
50     end
51     % Lettura di Coordinate di Fine Linea
52     % Definizione delle coordinate di Fine Linea
53     for j = 1:length(S_CPOS)
54         puntatore_end_CPOS = length(S_CPOS(j).X);
55         x_end_CAVI_POSITIVI = S_CPOS(j).X(puntatore_end_CPOS-1);
56         y_end_CAVI_POSITIVI = S_CPOS(j).Y(puntatore_end_CPOS-1);
57         tab_layer_CAVI_POSITIVI{j+1,3} = x_end_CAVI_POSITIVI;
58         tab_layer_CAVI_POSITIVI{j+1,4} = y_end_CAVI_POSITIVI;
59         x_end_CAVI_POSITIVI = 0;
60         y_end_CAVI_POSITIVI = 0;
61     end
62     tab_layer_CAVI_POSITIVI{1,1} = 'x_start';
63     tab_layer_CAVI_POSITIVI{1,2} = 'y_start';
64     tab_layer_CAVI_POSITIVI{1,3} = 'x_end';
65     tab_layer_CAVI_POSITIVI{1,4} = 'y_end';

```

5. Si commenta infine la parte di codice di importazione relativa al layer_LA analoga al ragionamento precedente:

```

1  % 4)Apertura del layer_LA e salvataggio
2  % Vettore: Colonne_layer_LA
3  % Colonna 1: LAYER
4  % Colonna 2: ATTRIBUTI
5  % Colonna 3: SEZIONE
6  % Colonna 4: CODICE UNIVOCO AUTOCAD
7  % Colonna 5: ORIGINE UNIVOCA AUTOCAD
8  % Colonna 6: LUNGHEZZA [m]
9  % Colonna 7: INIZIO
10 % Colonna 8: FINE
11 % Colonna 9: ID
12 % Lettura del file .csv ed importazione in tabella
    Colonne_layer_LA
13 % Apertura del File .csv

```

```
14     fid = fopen('layer_LA.csv');
15 % Lettura e scansione in tabella
16     Colonne_layer_LA = textscan(fid, '%S%S%S%S%S%S%S%S%S%S%S%S%S
    ','delimiter',' ');
17 % Chiusura del File .csv
18     fclose(fid);
19 % Lettura nome univoco e ID da tabella tramite iterazione di
    ciclo for
20 % apposito
21     name_LA = strcat(Colonne_layer_LA{4}, '-');
22     name_LA = strcat(name_LA, Colonne_layer_LA{1});
23     name_LA = strcat(name_LA, '-');
24     name_LA = strcat(name_LA, Colonne_layer_LA{9});
25     Num_LA = length(name_LA(:,1));
26 % Definizione dell'intestazione ed ID
27     for j=2:Num_LA
28         tab_layer_LA{j,6} = name_LA{j,1};
29     end
30     tab_layer_LA{1,6} = 'id';
31 % Calcolo della Resistenza
32 % Lettura Lunghezza
33     lunghezza_LA = str2double(Colonne_layer_LA{6});
34 % Inizializzazione variabile ausiliaria calcolo sezione
35     k = 0;
36 % Lettura Sezione
37     sezione_LA = str2double(Colonne_layer_LA{3});
38     for k = 2:Num_LA
39         if(strcmp(Colonne_layer_LA{1,3}{k,1}, 'LA')==1)
40             sezione_LA(k,1) = diametro_LA;
41         end
42     end
43 % Espressa in [ohm]
44     R_LA = rdivide(lunghezza_LA, sezione_LA) * 10^6 * pho;
45 % Definizione ed intestazione della Resistenza
46     for j=1:Num_LA
47         tab_layer_LA{j,5} = R_LA(j,1);
48     end
49     tab_layer_LA{1,5} = 'R';
50 % Lettura di Coordinate di Inizio Linea
51     [S_LA, A_LA] = shaperead('LA_GENERALE_linee.shp');
52 % Definizione delle coordinate di Inizio Linea
```

```

53     for j = 1:length(S_LA)
54         x_start_LA = S_LA(j).X(1);
55         y_start_LA = S_LA(j).Y(1);
56         tab_layer_LA{j+1,1} = x_start_LA;
57         tab_layer_LA{j+1,2} = y_start_LA;
58         x_start_LA = 0;
59         y_start_LA = 0;
60     end
61     % Lettura di Coordinate di Fine Linea
62     % Definizione delle coordinate di Fine Linea
63     for j = 1:length(S_LA)
64         puntatore_end_LA = length(S_LA(j).X);
65         x_end_LA = S_LA(j).X(puntatore_end_LA-1);
66         y_end_LA = S_LA(j).Y(puntatore_end_LA-1);
67         tab_layer_LA{j+1,3} = x_end_LA;
68         tab_layer_LA{j+1,4} = y_end_LA;
69         x_end_LA = 0;
70         y_end_LA = 0;
71     end
72     tab_layer_LA{1,1} = 'x_start';
73     tab_layer_LA{1,2} = 'y_start';
74     tab_layer_LA{1,3} = 'x_end';
75     tab_layer_LA{1,4} = 'y_end';

```

6. Conclusa la prima parte di importazione si ricostruisce la rete in maniera tale da velocizzare il processo di simulazione nel caso si vogliano effettuare modifiche sostanziali sulla configurazione dei tram e delle lame piuttosto che della rete. Questo ragionamento giustifica il codice con BASE e con TEST che distinguono rispettivamente la fase di simulazione della rete e la fase della configurazione della rete. Effettuo l'associazione di ogni cabina con i cavi positivi della rete in funzione di una tolleranza impostata dalle variabili globali. Applico questo ragionamento per la successiva associazione di un interruttore per ciascuna linea che parta dalla cabina configurabile dall'utenza.

```

1 % Codice BASE – Parte 2 – RICOSTRUZIONE DELLA RETE
2 % Tabelle Ordinate ottenute allo stato attuale
3 % 1)tab_layer_CASSETTE;
4 % 2)tab_layer_CAVI_POSITIVI;
5 % 3)tab_layer_CAVI_ALIMENTAZIONE;
6 % 4)tab_layer_LA;

```

```

7  % Ricavo punti di connessione per ciascuna cassetta collegata ad
    una cavo
8  % positivo con una certa tolleranza imposta
9  % Inizializzazione variabile ausiliaria puntatore tabella
10     j = 1;
11 % Inizializzazione variabile ausiliaria tabella confronto 1
12     k = 2;
13 %Inizializzazione variabile ausiliaria tabella confronto 2
14     w = 0;
15 %Flag di segnalazione errato aggancio tram per visualizzazione
16     flag_noTRAM = 0;
17 % Intestazione tabella cassette
18     tab_layer_CASSETTE{1,1} = 'x_start';
19     tab_layer_CASSETTE{1,2} = 'y_start';
20     tab_layer_CASSETTE{1,3} = 'x_end';
21     tab_layer_CASSETTE{1,4} = 'y_end';
22     tab_layer_CASSETTE{1,5} = 'R';
23     tab_layer_CASSETTE{1,6} = 'id';
24 % Intestazione lista connessioni cassette
25     sequenza_layer_CASSETTE{1,1} = 'x_start';
26     sequenza_layer_CASSETTE{1,2} = 'y_start';
27     sequenza_layer_CASSETTE{1,3} = 'x_end';
28     sequenza_layer_CASSETTE{1,4} = 'y_end';
29     sequenza_layer_CASSETTE{1,5} = 'R';
30     sequenza_layer_CASSETTE{1,6} = 'id';
31 % Definizione estremo ciclo in funzione del numero di elementi
    di gruppo CASSETTE
32     M = length(tab_layer_CASSETTE(:,1));
33     N = length(tab_layer_CAVI_POSITIVI(:,1));
34 % Controllero' tutte le coordinate x,y dei cavi positivi start
35     for w=2:M
36         target_newx = tab_layer_CASSETTE{w,1};
37         target_newy = tab_layer_CASSETTE{w,2};
38         % Inizializzazione variabile ausiliaria tabella
            confronto 1
39         k = 2;
40         while(k<=N)
41             varx = tab_layer_CAVI_POSITIVI{k,1};
42             vary = tab_layer_CAVI_POSITIVI{k,2};
43             % Se rientra nella tolleranza e' connesso

```

```

44         if(sqrt((target_newx-varx)^2 + (target_newy-vary
45             )^2)< tolleranza_cassetta)
46             j = j+1;
47             % Introduco il nodo fittizio di slack
48             sequenza_layer_CASSETTE{j,1} = 0;
49             % Introduco il nodo fittizio di slack
50             sequenza_layer_CASSETTE{j,2} = 0;
51             % Associo il centro della cassetta x di quel
52             determinato gruppo
53             sequenza_layer_CASSETTE{j,3} = target_newx;
54             % Associo il centro della cassetta y di quel
55             determinato gruppo
56             sequenza_layer_CASSETTE{j,4} = target_newy;
57             % Valore resistenza associata di gruppo (
58             trafo+convertitore)
59             sequenza_layer_CASSETTE{j,5} = R_CASSETTE;
60             sequenza_layer_CASSETTE{j,6} =
61             tab_layer_CASSETTE{w,6};
62             % Altrimenti se non rientro nella tolleranza
63             e non e' connesso
64         end
65         % Non ho una connessione con la cassetta passo ad un
66         'altra
67         % iterazione ma prima verifico ulteriormente non sia
68         magari
69         % una connessione con il punto end piuttosto che
70         start.
71         % Controllero' tutte le coordinate x,y dei cavi
72         positivi end
73     varx = tab_layer_CAVI_POSITIVI{k,3};
74     vary = tab_layer_CAVI_POSITIVI{k,4};
75     if(sqrt((target_newx-varx)^2 + (target_newy-vary
76         )^2)< tolleranza_cassetta)
77         j = j+1;
78         % Introduco il nodo fittizio di slack
79         sequenza_layer_CASSETTE{j,1} = 0;
80         % Introduco il nodo fittizio di slack
81         sequenza_layer_CASSETTE{j,2} = 0;
82         % Associo il centro della cassetta x di quel
83         determinato gruppo
84         sequenza_layer_CASSETTE{j,3} = target_newx;

```

```

73         % Associo il centro della cassetta y di quel
           determinato gruppo
74         sequenza_layer_CASSETTE{j,4} = target_newy;
75         % Valore resistenza associata di gruppo (
           trafo+convertitore)
76         sequenza_layer_CASSETTE{j,5} = R_CASSETTE;
77         sequenza_layer_CASSETTE{j,6} =
           tab_layer_CASSETTE{w,6};
78         end
79         % Inizializzazione variabile ausiliaria tabella
           confronto 1
80         k = k+1;
81     end
82 end

```

7. Si effettua il calcolo della matrice dei nodi confrontando le coordinate ricavate di tutti gli elementi della rete.

```

1  % Codice BASE – Parte 3 – PROCEDURA DI CALCOLO AUTOMATICO
2  % Generazione della tabella generale
3      tab_layer_CAVI_POSITIVI(1,:) = [];
4      tab_layer_CAVI_ALIMENTAZIONE(1,:) = [];
5      tab_layer_LA(1,:) = [];
6      tab_generale = [sequenza_layer_CASSETTE;
7                      tab_layer_CAVI_POSITIVI; tab_layer_CAVI_ALIMENTAZIONE;
8                      tab_layer_LA];
9  % Generazione della matrice delle incidenze
10 % Inizializzazione variabile contatore 1
11     i = 0;
12 % Inizializzazione variabile contatore 2
13     j = 0;
14 % Inizializzazione flag pari a zero
15     flag = 0;
16 % Inizializzazione contatore indice matrice_nodi
17     w_nodo = 1;
18 % Inizializzazione contatore ciclo matrice_nodi
19     k = 0;
20 % Intestazione della matrice dei nodi
21     matrice_nodi{1,1} = 'X';
22     matrice_nodi{1,2} = 'Y';
23     matrice_nodi{1,3} = 'id';

```

```
22 % Calcolo della lunghezza della tabella generale
23     N = length(tab_generale);
24 % Calcolo della lunghezza della matrice_nodi
25     M = length(matrice_nodi);
26     for i=2:N
27         % Confronto la colonna START con se' stessa
28             for j=2:N
29                 % Condizione che mi verifica elementi non ripetibili
30                 % Prendo il primo elemento lo confronto con altri e se
31                 % trovo
32                 % coincidenza ignoro se e' nuovo lo aggiungo alla lista
33                 % verificando
34                 % di non immettere gli stessi valori gia' riscontrati
35                 % precedentemente
36                 % Inizializzazione verifica ad ogni iterazione
37                 flag = 0;
38                 % Calcolo della lunghezza della matrice_nodi
39                 % variabile ogni ciclo
40                 M = length(matrice_nodi);
41                 % Non possono coincidere righe della stessa colonna
42                 % e verifico se sono piene
43                 if(isequal(i,j) == 0 && isempty(tab_generale{i,1})
44                 == 0 && isempty(tab_generale{i,2})== 0)
45                     if(abs(tab_generale{i,1}-tab_generale{j,1}) >
46                     epsilon)
47                         if(abs(tab_generale{i,2}-tab_generale{j,2})
48                         > epsilon)
49                             % Verificato che la prima colonna ha
50                             % valore univoco
51                             % differente prendo questo valore e se
52                             % passa il
53                             % test di check della matrice colonne e'
54                             % ok e lo
55                             % posso inserire (non ho copioni sulla
56                             % matrice_nodi
57                             % flag = 0, altrimenti flag = 1)
58                             % Significa che la matrice_nodi e' piena
59                             % e posso iniziare a confrontarla
60                             if(w_nodo>2)
61                                 % Test di check con tolleranza
62                                 imposta non
```

```
49         % modificabile come variabile
          globale pari
50         % a epsilon da prove sperimentali
51         for k = 2:M
52             if(abs(tab_generale{i,1}-
                    matrice_nodi{k,1})< epsilon)
53                 if(abs(tab_generale{i,2}-
                    matrice_nodi{k,2})<
                    epsilon)
54                     % Significa che ha
                        trovato un doppione e
                        non aggiorna in
                        tabella
55                     flag = 1;
56                 end
57             end
58         end
59     end
60     % Al primo ciclo for l'iterazione
        entrera' qua dentro
61     if (flag == 0)
62         % Condizione di verifica per w = 2
63         if(w_nodo==3)
64             if(abs(matrice_nodi{k,1}-
                    matrice_nodi{k-1,1})< epsilon
                )
65                 if(abs(matrice_nodi{k,2}-
                    matrice_nodi{k-1,2})<
                    epsilon)
66                     % Decremento contatore
                        indice matrice_nodi
67                     w_nodo = w_nodo-1;
68                 end
69             end
70         end
71         %Incremento del contatore indice
            matrice_nodi
72         w_nodo = w_nodo+1;
73         % Aggiornamento variabili in
            matrice_nodi
74         matrice_nodi{w_nodo,3} = w_nodo-1;
```

```
75         matrice_nodi{w_nodo,1} = tab_generale{i
           ,1};
76         matrice_nodi{w_nodo,2} = tab_generale{i
           ,2};
77         end
78     end
79 end
80 end
81 end
82 end
83 % Inizializzazione variabile contatore 1
84     i = 0;
85 % Inizializzazione variabile contatore 2
86     j = 0;
87 % Inizializzazione flag pari a zero
88     flag = 0;
89 % Inizializzazione contatore ciclo matrice_nodi
90     k = 0;
91 % Calcolo della lunghezza della tabella generale
92     N = length(tab_generale);
93 % Calcolo della lunghezza della matrice_nodi
94     M = length(matrice_nodi);
95     for i=2:N
96         % Confronto la colonna END con la matrice_nodi
97         for j=2:N
98             % Condizione che mi verifica elementi non ripetibili
99             % Prendo il primo elemento lo confronto con altri e se
           trovo
100             % coincidenza ignoro se e' nuovo lo aggiungo alla lista
           verificando
101             % di non immettere gli stessi valori gia' riscontrati
           precedentemente
102             % Inizializzazione verifica ad ogni iterazione
103             flag = 0;
104             % Calcolo della lunghezza della matrice_nodi
           variabile ogni ciclo
105             M = length(matrice_nodi);
106             % Non possono coincidere righe della stessa colonna
           e verifico se sono piene
107             if(isequal(i,j) == 0 && isempty(tab_generale{i,3})
               == 0 && isempty(tab_generale{i,4})== 0)
```

```
108     if(abs(tab_generale{i,3}-tab_generale{j,3}) >
        epsilon)
109         if(abs(tab_generale{i,4}-tab_generale{j,4})
            > epsilon)
110             % Verificato che la prima colonna ha
                valore univoco
111             % differente prendo questo valore e se
                passa il
112             % test di check della matrice colonne e'
                ok e lo
113             % posso inserire (non ho copioni sulla
                matrice_nodi
114             % flag = 0, altrimenti flag = 1)
115             % Significa che la matrice_nodi e' piena
                e posso iniziare a confrontarla
116             if(w_nodo>2)
117                 % Test di check con tolleranza
                    imposta
118                 for k = 2:M
119                     if(abs(tab_generale{i,3}-
                        matrice_nodi{k,1})< epsilon)
120                         if(abs(tab_generale{i,4}-
                            matrice_nodi{k,2})<
                                epsilon)
121                             % Significa che ha
                                    trovato un doppione e
                                    non aggiorna in
                                    tabella
122                             flag = 1;
123                             end
124                         end
125                     end
126                 end
127             % Al primo ciclo for l'iterazione
                entrera' qua dentro se e' ok aggiorna
                tabelle
128             if (flag == 0)
129                 % Condizione di verifica per w = 2
130                 if(w_nodo==3)
131                     if(abs(matrice_nodi{k,1}-
                        matrice_nodi{k-1,1})< epsilon)
```

```

132         if(abs(matrice_nodi{k,2}-
133             matrice_nodi{k-1,2})<
134             epsilon)
135             % Decremento contatore
136             indice matrice_nodi
137             w_nodo = w_nodo-1;
138         end
139     end
140 end
141 % Incremento del contatore indice
142 matrice_nodi
143 w_nodo = w_nodo+1;
144 % Aggiornamento variabili in
145 matrice_nodi
146 matrice_nodi{w_nodo,3} = w_nodo-1;
147 matrice_nodi{w_nodo,1} = tab_generale{i
148     ,3};
149 matrice_nodi{w_nodo,2} = tab_generale{i
150     ,4};
151 end
152 end
153 end
154 end
155 end
156 % Fine della fase di pre-caricamento delle variabili di sistema
157 % automaticamente ho il salvataggio delle variabili di interesse
158 % e avvio
159 % della fase di test.
160 save('variabili_pre_caricate');
161 % Avvio della fase di Test
162 run('AGT_TEST_1V.m');

```

8. Si commenta la sezione di codice della parte relativa al layer_LAME:

```

1 % Codice Tesi Magistrale Bertolone Citin Lorenzo
2 % CODICE DI TEST
3 % Licenza Creative Commons – Attribuzione
4 % CC BY Bertolone Citin Lorenzo
5 % Progetto AUGUSTA TAURINGRID
6 % AGT_TEST_1V.m

```

```
7 % COMMENTI DI PROCEDURA
8   clear; %Pulisce le variabili di sistema
9   close all;%Chiude tutte le finestre aperte
10  clc; %Pulisce la command window di MATLAB
11  format long; %Imposta visibilita' dei risultati piu'
    accurata
12  load('variabili_pre_caricate'); %Carica il file pre-
    impostato di rete contenente le variabili
13 % Codice TEST – Parte 1 – IMPORTAZIONE DEI DATI DA QGIS
14 % PARTE DI CODICE DA SVOLGERE SUCCESSIVAMENTE AL PRE-CARICAMENTO
15 % 5)Apertura del layer_LAME e salvataggio
16 % Vettore: Colonne_layer_LAME
17 % Colonna 1: LAYER
18 % Colonna 2: ATTRIBUTI
19 % Colonna 3: CODICE UNIVOCO AUTOCAD
20 % Colonna 4: ORIGINE UNIVOCA AUTOCAD
21 % Colonna 5: LUNGHEZZA
22 % Colonna 6: SEQUENZA
23 % Colonna 7: ID
24 % Lettura del file .csv ed importazione in tabella
    Colonne_layer_LAME
25 % Apertura del File .csv
26   fid = fopen('layer_LAME.csv');
27 % Lettura e scansione in tabella
28   Colonne_layer_LAME = textscan(fid, '%s%s%s%s%s%s', '
    delimiter','\n');
29 % Chiusura del File .csv
30   fclose(fid);
31 % La pulizia viene effettuata per estrapolare i dati necessari
    ed
32 % alleggerire il peso dei dati contenuti nelle tabelle
    attraverso apposite
33 % operazioni di sostituzione e delimitazione di stringa
34 % Inizializzazione variabile ausiliaria lettura dato che ogni
    colonna contiene un numero indefinito di righe
35   t = 0;
36 % Inizializzazione indice ausiliario lettura dati
37   k = 0;
38   for s=1:length(Colonne_layer_LAME(1,:))
39     % Alla fine della lettura della serie di righe in una
        colonna riprendi da capo la colonna successiva
```

```

40     t = 0;
41     while(t<length(Colonne_layer_LAME{: ,s}))
42         t = t+1;
43         k = k+1;
44         % Nel codice originale presenza delle doppie
           virgolette
45         Lista_layer_LAME{k,1} = strrep(Colonne_layer_LAME{s
           }{t}, ``MultiPoint ((' , '));
46         Lista_layer_LAME{k} = strrep(Lista_layer_LAME{k}, ' , '
           ');
47         Lista_layer_LAME{k} = strrep(Lista_layer_LAME{k}, ' ( ' , ' '
           ');
48         % Nel codice originale presenza delle doppie
           virgolette
49         Lista_layer_LAME{k} = strrep(Lista_layer_LAME{k}, '))
           `` , ' ');
50         % Strrep permette di sostituire parti di stringa con
           altre
51         % Nel codice originale presenza delle doppie
           virgolette
52         Lista_layer_LAME{k} = strrep(Lista_layer_LAME{k}, ``
           ' , ' ');
53         % Regexp permette di differenziare tra loro elementi
           sulla base di un'altra stringa
54         Lista_layer_LAME{k} = regexp(Lista_layer_LAME{k}, ' '
           , 'split');
55     end
56 end
57 % Variabile ausiliaria di verifica dove uno negativo giustifica
           la prima riga di intestazione
58     Num_Lame = k-1;
59 % Sulla base della tabella Lista_layer_LAME trasferisco tutti i
           parametri di interesse in tab_layer_LAME tramite iterazione di
60 % apposito ciclo for
61 % Variabili di inizializzazione del ciclo for
62 % Inizializzazione variabile che conteggia il numero di riga
           della tabella
63     k = 2;
64 % Inizializzazione variabile che differenzia colonna x e y della
           tabella
65     j = 1;
66

```

```
67 % Inizializzazione variabile ausiliaria di tabella utilizzata
    come indice puntatore
68     w = 2;
69 % Inizializzazione ed intestazione della tabella
70     tab_sequenza_layer_LAME{1,1} = 'IDENTIFICAZIONE';
71     tab_sequenza_layer_LAME{1,2} = 'VERIFICA NODO';
72     tab_sequenza_layer_LAME{1,3} = 'x';
73     tab_sequenza_layer_LAME{1,4} = 'y';
74     tab_sequenza_layer_LAME{1,5} = 'NODO_RIF';
75     tab_sequenza_layer_LAME{1,6} = 'id';
76 % Ridefinizione delle coordinate da shapefile per lettura
    corretta del
77 % vettore multipoint
78     [S_LAME,A_LAME] = shaperead('LAME_linee.shp');
79 % Estrapolo i dati relativi alla sequenza ordinata di coordinate
    della
80 % struttura LAME
81 % Contatore ausiliario riferimento nodo lame
82     s=0;
83 % Contatore per verifica lame
84     t=0;
85 % Variabile ausiliaria di segnalazione
86     flag = 0;
87     z = 0;
88     r = 0;
89 % Condizione tale per cui il ciclo while continuerà iterazione
    con variabile che conteggia il numero di riga della tabella
90     while(k <= Num_Lame+1)
91         % Inizializzazione variabile che differenzia colonna x e y
            della tabella
92         j = 6;
93         r = 0;
94         while(j < length(Lista_layer_LAME{k,1}))
95             % Condizione tale per cui il ciclo while continuerà
                iterazione con variabile che differenzia
                colonna x e y della tabella
96             % Concateno numero identificativo univoco e ID della
                lama
97             tab_sequenza_layer_LAME{w,6} = strcat(
                Lista_layer_LAME{k,1}{1,3}, '-LM-');
```

```
98     tab_sequenza_layer_LAME{w,6} = strcat(  
        tab_sequenza_layer_LAME{w,6},Lista_layer_LAME{k  
        ,1}{1,length(Lista_layer_LAME{k,1})});  
99     % L'id parte da 0;  
100    lame_gruppi_id = str2double(Lista_layer_LAME{k,1}{1,  
        length(Lista_layer_LAME{k,1})});  
101    % Conversione in double da stringa ed  
        identificazione delle coordinate x,y  
102    r = r+1;  
103    x_sequenza_layer_LAME = S_LAME(lame_gruppi_id+1).X(r  
        );  
104    y_sequenza_layer_LAME = S_LAME(lame_gruppi_id+1).Y(r  
        );  
105    tab_sequenza_layer_LAME{w,3} = x_sequenza_layer_LAME  
        ;  
106    tab_sequenza_layer_LAME{w,4} = y_sequenza_layer_LAME  
        ;  
107    tab_sequenza_layer_LAME{w,1} = 'LAME';  
108    x_sequenza_layer_LAME = 0;  
109    y_sequenza_layer_LAME = 0;  
110    % Conteggio per ogni lama il numero identificativo  
        ordinato di nodi  
111    % Questo ciclo e' lo step successivo a quello  
        iniziale con flag pari a  
112    % zero  
113    % Alla prima iterazione non parte perche' il flag e'  
        inizialmente nullo  
114    if (flag == 1)  
115    % Lettura univoca del gruppo di lame associato  
116    var_lame{w,1} = lame_gruppi_id;  
117    % Condizione tale per cui rimango sullo stesso  
        gruppo lame  
118    if(var_lame{w-1,1} == var_lame{w,1})  
119        % Incremento contatore ausiliario riferimento  
        nodo lame  
120        s=s+1;  
121        % Scrittura contatore ausiliario riferimento  
        nodo lame su tabella  
122        tab_sequenza_layer_LAME{w,5} = s;  
123        % Incremento contatore per verifica nodo lame  
124        t = t+1;
```

```
125     else
126         % Altrimenti se cambio gruppo lame resetto
           conattore ausiliario riferimento nodo
127         s = 1;
128         % Scrittura contatore ausiliario riferimento
           nodo lame su tabella
129         tab_sequenza_layer_LAME{w,5} = s;
130         % Con almeno 3 punti di connessioni e' un nodo
           altrimenti l'esito della verifica e' negativo
131         if(t>2)
132             while (t~=0)
133                 % Esito verifica positivo su tabella per
                   tutti gli elementi del gruppo
134                 tab_sequenza_layer_LAME{w-t,2} = 1;
135                 % Decremento contatore per inserire a
                   tabella esito per ciascun elemento
                   del gruppo
136                 t = t-1;
137             end
138         end
139         % Con 2 punti di connessione non e' un nodo
140         if(t<=2)
141             while (t~=0)
142                 % Esito verifica negativo su tabella per
                   tutti gli elementi del gruppo
143                 tab_sequenza_layer_LAME{w-t,2} = 0;
144                 % Decremento contatore per inserire a
                   tabella esito per ciascun elemento
                   del gruppo
145                 t = t-1;
146             end
147         end
148         % Reset contatore verifica nodo lame
149         t = 1;
150     end
151 end
152 % Questo ciclo e' lo step iniziale
153 % Alla prima iterazione inizio da qui dato che impongo flag
           inizialmente nullo
154 if (flag == 0)
155     % Contatore per verifica nodo lame inizializzazione
```

```
156     t = t+1;
157     % Contatore contatore ausiliario riferimento nodo lame
        inizializzazione
158     s = s+1;
159     % Lettura univoca del gruppo di lame associato
160     var_lame{w,1} = lame_gruppi_id;
161     % Scrittura contatore ausiliario riferimento nodo lame
        su tabella
162     tab_sequenza_layer_LAME{w,5} = s;
163     % Imposizione flag pari a uno per successivi cicli
164     flag = 1;
165     end
166     % Incremento doppio variabile che differenzia colonna x e y
        della tabella
167     j = j+2;
168     % Incremento variabile ausiliaria di tabella utilizzata come
        indice puntatore
169     w = w+1;
170     end
171     % Uscendo dal ciclo devo concludere il calcolo per l'ultima
        iterazione
172     % risolvendo i calcoli in maniera esattamente uguale a
        quando ero
173     % all'interno del ciclo
174     % Con almeno 3 punti di connessioni e' un nodo altrimenti l'
        esito della verifica e' negativo
175         if(t>2)
176             while (t~=0)
177                 % Esito verifica positivo su tabella per
                    tutti gli elementi del gruppo
178                 tab_sequenza_layer_LAME{w-t,2} = 1;
179                 % Decremento contatore per inserire a
                    tabella esito per ciascun elemento del
                    gruppo
180                 t = t-1;
181             end
182         end
183     % Con 2 punti di connessione non e' un nodo
184     if(t<=2)
185         while (t~=0)
```

```

186         % Esito verifica negativo su tabella per tutti
           gli elementi del gruppo
187         tab_sequenza_layer_LAME{w-t,2} = 0;
188         % Decremento contatore per inserire a tabella
           esito per ciascun elemento del gruppo
189         t = t-1;
190         end
191     end
192     %Reset contatore verifica nodo lame
193     t = 0;
194     k = k+1;
195 end

```

9. Si commenta la sezione di codice della parte 2 relativa al layer_TRAM dove ricerco la posizione del tram sulla linea aerea attraverso il seguente algoritmo:

```

1  % 6)Apertura del layer_TRAM e salvataggio dei dati
2  % Vettore: Colonne_layer_TRAM
3  % Colonna 1: id
4  % Colonna 2: X
5  % Colonna 3: Y
6  % Serie di colonne Join di associazione alla linea da 4 a 18
7  % Lettura del file .csv ed importazione in tabella
   Colonne_layer_TRAM
8  % Apertura del File .csv
9  fid = fopen('layer_TRAM.csv');
10 % Lettura e scansione in tabella
11 Colonne_layer_TRAM_punti = textscan(fid,'%s%s%s%s%s%s%s%s%s%
   s%s%s%s%s%s%s%s', 'delimiter', ',');
12 % Chiusura del File .csv
13 fclose(fid);
14 % Lettura nome univoco ID e posizione in coordinate:
15 % Coordinate x ed intestazione ID
16 x_TRAM = Colonne_layer_TRAM_punti{2};
17 TIPO_TRAM = Colonne_layer_TRAM_punti{4};
18 for j=2:length(x_TRAM)
19     if(isequal(x_TRAM, '')==0) %Esclude tutti i valori nulli
20         tab_layer_TRAM{j,1} = str2double(x_TRAM(j,1));
21         tab_layer_TRAM{j,3} = 0;
22         tab_layer_TRAM{j,4} = 0;

```

```
23 % Potrei ipotizzare modello 1 motrice serie 2800,  
    modello 2  
24 % motrice serie 5000, modello 3 motrice serie 6000  
    in  
25 % accelerazione o in frenata con la convenzione A/F  
26 if(isequal(TIPO_TRAM{j,1}, '28001')==1)  
27 % In frenata  
28 % Parametro variabile da tabella QGIS  
29     G_TRAM = 10e-06; % Serie 2800  
30     I_TRAM = -1000; % Serie 2800  
31     tab_layer_TRAM{j,5} = 1/G_TRAM;  
32 end  
33 % Ipotizzo in fase iniziale serie 2800 in  
    accelerazione  
34 % Tipologia modello esemplificato da 1000[A] a 600[V  
    ]  
35 if(isequal(TIPO_TRAM{j,1}, '28002')==1)  
36 % In accelerazione  
37 % Variabile di sistema modificabile in funzione del  
    .csv  
38     G_TRAM = 10e-06; % Serie 2800  
39     I_TRAM = 1000; % Serie 2800  
40     tab_layer_TRAM{j,5} = 1/G_TRAM;  
41 end  
42 if(isequal(TIPO_TRAM{j,1}, '50001')==1)  
43 % In frenata  
44 % Variabile di sistema modificabile in funzione del  
    .csv  
45     G_TRAM = 10e-06; % Serie 5000  
46     I_TRAM = -1000; % Serie 5000  
47     tab_layer_TRAM{j,5} = 1/G_TRAM;  
48 end  
49 if(isequal(TIPO_TRAM{j,1}, '50002')==1)  
50 % Variabile di sistema modificabile in funzione del  
    .csv  
51 % In accelerazione  
52     G_TRAM = 10e-06; % Serie 5000  
53     I_TRAM = 1000; % Serie 5000  
54     tab_layer_TRAM{j,5} = 1/G_TRAM;  
55 end  
56 if(isequal(TIPO_TRAM{j,1}, '60001')==1)
```

```

57     % Variabile di sistema modificabile in funzione del
        .csv
58     % In frenata
59         G_TRAM = 10e-06; % Serie 6000
60         I_TRAM = -1000; % Serie 6000
61         tab_layer_TRAM{j,5} = 1/G_TRAM;
62     end
63     if(isequal(TIPO_TRAM{j,1}, '60002')==1)
64     % Variabile di sistema modificabile in funzione del
        .csv
65     % In accelerazione
66         G_TRAM = 10e-06; % Serie 6000
67         I_TRAM = 1000; % Serie 6000
68         tab_layer_TRAM{j,5} = 1/G_TRAM;
69     end
70     % Ipotesi di Guasto
71     if(isequal(TIPO_TRAM{j,1}, '0')==1)
72     % Variabile di sistema modificabile in funzione del
        .csv
73     % In condizioni di guasto franco
74         G_TRAM = 10e06;
75         I_TRAM = 0;
76         tab_layer_TRAM{j,5} = 1/G_TRAM;
77     end
78     end
79     end
80     tab_layer_TRAM{1,1} = 'x';
81     % Coordinate y ed intestazione ID
82     y_TRAM = Colonne_layer_TRAM_punti{3};
83     % Pulizia Stringhe ed iterazione tramite apposito ciclo for
84     Colonne_layer_TRAM_punti{1} = strjoin(
        Colonne_layer_TRAM_punti{1});
85     Colonne_layer_TRAM_punti{1} = regexp(
        Colonne_layer_TRAM_punti{1}, ' ', 'split');
86     for j=2:length(y_TRAM)
87         if(isequal(y_TRAM, '')==0) %Esclude tutti i valori nulli
88             tab_layer_TRAM{j,2} = str2double(y_TRAM(j,1));
89             name_TRAM = strcat('TR-',Colonne_layer_TRAM_punti
                {1,1}{1,j});
90             tab_layer_TRAM{j,6} = name_TRAM;
91     end

```

```

92     end
93     tab_layer_TRAM{1,2} = 'y';
94     tab_layer_TRAM{1,5} = 'R_TRAM';
95     tab_layer_TRAM{1,6} = 'id';
96     tab_layer_TRAM{1,7} = 'X_START_LA';
97     tab_layer_TRAM{1,8} = 'Y_START_LA';
98     tab_layer_TRAM{1,9} = 'X_END_LA';
99     tab_layer_TRAM{1,10} = 'Y_END_LA';
100 % Lettura di Coordinate di Inizio Linea
101 % Pulizia Stringhe ed iterazione tramite apposito ciclo for
102     start_TRAM = strrep(Colonne_layer_TRAM_punti{11}, 'Point (', '
103     ');
104     Colonne_layer_TRAM_punti{11} = strrep(start_TRAM, ')', '');
105     start_TRAM = regexp(Colonne_layer_TRAM_punti{11}, ' ', 'split'
106     );
107 % Definizione delle coordinate di Inizio Linea
108     for j = 2:length(start_TRAM)
109         x_start_TRAM = str2double(start_TRAM{j,1}{1,1});
110         y_start_TRAM = str2double(start_TRAM{j,1}{1,2});
111         tab_layer_TRAM{j,7} = x_start_TRAM;
112         tab_layer_TRAM{j,8} = y_start_TRAM;
113         x_start_TRAM = 0;
114         y_start_TRAM = 0;
115     end
116 % Lettura di Coordinate di Fine Linea
117 % Pulizia Stringhe ed iterazione tramite apposito ciclo for
118     end_TRAM = strrep(Colonne_layer_TRAM_punti{12}, 'Point (', '')
119     ;
120     Colonne_layer_TRAM_punti{12} = strrep(end_TRAM, ')', '');
121     end_TRAM = regexp(Colonne_layer_TRAM_punti{12}, ' ', 'split');
122 % Definizione delle coordinate di Fine Linea
123     for j = 2:length(end_TRAM)
124         x_end_TRAM = str2double(end_TRAM{j,1}{1,1});
125         y_end_TRAM = str2double(end_TRAM{j,1}{1,2});
126         tab_layer_TRAM{j,9} = x_end_TRAM;
127         tab_layer_TRAM{j,10} = y_end_TRAM;
128         x_end_TRAM = 0;
129         y_end_TRAM = 0;
130     end
131 % Ricavo la posizione relativa del tram agli estremi della linea
132     aerea

```

```
129 % tramite un algoritmo che preveda l'associazione degli estremi
    con
130 % iterazioni multiple
131 % Definizione estremo ciclo in funzione del numero di elementi
    di gruppo TRAM
132     M = length(tab_layer_TRAM(:,1));
133 % Inizializzazione flag pari a zero
134     flag = 0;
135     for w=2:M
136         % Le coordinate del TRAM sono:
137         % target_x
138         % target_y
139         target_x = tab_layer_TRAM{w,1};
140         target_y = tab_layer_TRAM{w,2};
141         flag = 0;
142         % Inizializzazione flag pari a zero ad ogni iterazione
143         % Confronto con il punto di start
144         varx_start = tab_layer_TRAM{w,7};
145         vary_start = tab_layer_TRAM{w,8};
146         distanza_start = sqrt((target_x-varx_start)^2 + (
            target_y-vary_start)^2);
147         % Confronto con il punto di end
148         varx_end = tab_layer_TRAM{w,9};
149         vary_end = tab_layer_TRAM{w,10};
150         distanza_end = sqrt((target_x-varx_end)^2 + (
            target_y-vary_end)^2);
151         %Il tram fa contatto nell'estremo iniziale
152         if(distanza_start < distanza_end)
153             varx_tram_0 = varx_start; %Refresh variabili
154             vary_tram_0 = vary_start; %Refresh variabili
155             flag = 1; %Flag pari a uno per ignorare le altre
                opzioni
156         end
157         if(distanza_start > distanza_end) %Il tram fa
            contatto nell'estremo finale
158             varx_tram_0 = varx_end; %Refresh variabili
159             vary_tram_0 = vary_end; %Refresh variabili
160             flag = 1; %Flag pari a uno per ignorare le altre
                opzioni
161         end
162         % Se le 2 distanze sono uguali
```

```

163         if(distanza_start == distanza_end) %Il tram fa
           contatto nell'estremo finale per convenzione
164             varx_tram_0 = varx_end; %Refresh variabili
165             vary_tram_0 = vary_end; %Refresh variabili
166             flag = 1; %Flag pari a uno per ignorare le altre
               opzioni
167         end
168     if (flag == 1) %Flag pari a uno per non effettuare il
       refresh
169         tab_layer_TRAM{w,1} = 0;
170         tab_layer_TRAM{w,2} = 0;
171         tab_layer_TRAM{w,3} = varx_tram_0; %Mantiene valori
           iterazione precedente
172         tab_layer_TRAM{w,4} = vary_tram_0; %Mantiene valori
           iterazione precedente
173         flag_noTRAM = 0;
174     end
175     % Segnalazione di errore e stampa su tabella se non trova
       con tolleranza il tram e flag = 0
176     if (flag == 0)
177         tab_layer_TRAM{w,3} = 'NON RILEVATO';
178         tab_layer_TRAM{w,4} = 'NON RILEVATO';
179         flag_noTRAM = 1;
180     end
181 end

```

10. Analizzo la parte di codice relativa alla ricostruzione dei lati delle lame.

```

1 % Codice TEST – Parte 2 – RICOSTRUZIONE DELLA RETE
2 % Analizzo le lame e nello specifico impongo dei parametri di
3 % idealizzazione dato che la loro lunghezza e' ritenuta
   trascurabile
4 % rispetto ai fini pratici confrontati con la lunghezza
   complessiva della
5 % rete
6 % Ipotizzo per le lame sezione 95[mm^2] e lunghezza 20[cm]
   ovvero 0.2[m]
7 % Intestazione tabella cassette
8     tab_layer_LAME{1,1} = 'x_start';
9     tab_layer_LAME{1,2} = 'y_start';
10    tab_layer_LAME{1,3} = 'x_end';

```

```

11 tab_layer_LAME{1,4} = 'y_end';
12 tab_layer_LAME{1,5} = 'R';
13 tab_layer_LAME{1,6} = 'id';
14 % Resistenza delle lame
15 R_LAME = rdivide(lunghezza_LAME,sezione_LAME)* 10^6 * pho;
16 % Inizializzazione variabile ausiliaria puntatore tabella
17 j = 1;
18 % Inizializzazione flag pari a zero
19 flag = 0;
20 % Inizializzazione variabile ausiliaria tabella confronto 1
21 k = 0;
22 % Inizializzazione variabile ausiliaria tabella confronto 2
23 w = 0;
24 % Definizione estremo ciclo in funzione del numero di
    elementi di gruppo LAME
25 M = length(tab_sequenza_layer_LAME(:,1))-1;
26 for k=2:M
27 % Variabile ausiliaria tabella confronto 2 incrementata di 1
    rispetto alla variabile ausiliaria tabella confronto 1
28 w=k+1;
29 % Non e' la prima iterazione dato che flag e'
    inizializzato a zero
30 if (flag ==1)
31     if(tab_sequenza_layer_LAME{w,5}~=1 && w~=M)
32         % Incremento variabile ausiliaria puntatore
            tabella e aggiorno le variabili
33         j = j+1;
34         tab_layer_LAME{j,1} = tab_sequenza_layer_LAME{k
            ,3};
35         tab_layer_LAME{j,2} = tab_sequenza_layer_LAME{k
            ,4};
36         tab_layer_LAME{j,6} = tab_sequenza_layer_LAME{k
            ,6};
37         tab_layer_LAME{j,3} = tab_sequenza_layer_LAME{w
            ,3};
38         tab_layer_LAME{j,4} = tab_sequenza_layer_LAME{w
            ,4};
39         tab_layer_LAME{j,5} = R_LAME;
40     end
41 end
42 if(flag ==0)

```

```

43     % E' la prima iterazione dato che flag e'
        inizializzato a zero
44     % Incremento variabile ausiliaria puntatore
        tabella e aggiorni le variabili
45     j = j+1;
46     tab_layer_LAME{j,1} = tab_sequenza_layer_LAME{k
        ,3};
47     tab_layer_LAME{j,2} = tab_sequenza_layer_LAME{k
        ,4};
48     tab_layer_LAME{j,6} = tab_sequenza_layer_LAME{k
        ,6};
49     tab_layer_LAME{j,3} = tab_sequenza_layer_LAME{w
        ,3};
50     tab_layer_LAME{j,4} = tab_sequenza_layer_LAME{w
        ,4};
51     tab_layer_LAME{j,5} = R_LAME;
52     flag = 1;
53     % Inizializzazione flag pari a uno
54     end
55     if(w==M)
56         % E' sicuramente l'ultimo dato che lame da un
            singolo
57         % vertice sono inesistenti e non faccio nulla
58     end
59     %Se tab_sequenza_layer_LAME{w,5} = 1, allora fara' N-1
        rami rispetto al numero di nodi
60 end

```

11. Esamino la parte di codice che si occupa di elaborare la matrice nodi già esistente e completarla in funzione della configurazione di rete:

```

1 % Codice TEST – Parte 3 – PROCEDURA DI CALCOLO AUTOMATICO
2 % Generazione della tabella generale
3     tab_layer_LAME(1,:) = [];
4     tab_layer_TRAM(1,:) = [];
5     tab_generale_new = [tab_layer_LAME; tab_layer_TRAM(:,(1:6))
        ];
6 % Generazione della matrice delle incidenze
7 % Inizializzazione variabile contatore 1
8     i = 0;
9 % Inizializzazione variabile contatore 2

```

```
10     j = 0;
11 % Inizializzazione contatore ciclo matrice_nodi
12     k = 0;
13 % Calcolo della lunghezza della tabella generale
14     N = length(tab_generale_new);
15 % Calcolo della lunghezza della matrice_nodi
16     M = length(matrice_nodi);
17     for i=1:N
18         % Confronto la colonna START con se' stessa
19         for j=1:N
20             % Condizione che mi verifica elementi non ripetibili
21             % Prendo il primo elemento lo confronto con altri e se
                trovo
22             % coincidenza ignoro se e' nuovo lo aggiungo alla lista
                verificando
23             % di non immettere gli stessi valori gia' riscontrati
                precedentemente
24             % Inizializzazione verifica ad ogni iterazione
25             flag = 0;
26             % Calcolo della lunghezza della matrice_nodi
                variabile ogni ciclo
27             M = length(matrice_nodi);
28             if(isequal(i,j) == 0 && isempty(tab_generale_new{i
                ,1}) == 0 && isempty(tab_generale_new{i,2})== 0)
29                 % Non possono coincidere righe della stessa colonna
                e verifico se sono piene
30                 if(abs(tab_generale_new{i,1}-tab_generale_new{j
                ,1}) > epsilon)
31                     if(abs(tab_generale_new{i,2}-
                tab_generale_new{j,2}) > epsilon)
32                         % Verificato che la prima colonna ha
                valore univoco
33                         % differente prendo questo valore e se
                passa il
34                         % test di check della matrice colonne e'
                ok e lo
35                         % posso inserire (non ho copioni sulla
                matrice_nodi
36                         % flag = 0, altrimenti flag = 1)
37                         % Significa che la matrice_nodi e' piena
                e posso iniziare a confrontarla
```

```
38         if(w_nodo>2)
39             % Test di check con tolleranza
40             imposta non
41             % modificabile come variabile
42             globale pari
43             % a epsilon da prove sperimentali
44             for k = 2:M
45                 if(abs(tab_generale_new{i,1}-
46                     matrice_nodi{k,1})< epsilon)
47                     if(abs(tab_generale_new{i
48                         ,2}-matrice_nodi{k,2})<
49                         epsilon)
50                         % Significa che ha trovato
51                         un doppione e non
52                         aggiorna in tabella
53                         flag = 1;
54                     end
55                 end
56             end
57         end
58         if (flag == 0)
59             % Incremento del contatore indice
60             matrice_nodi
61             w_nodo = w_nodo+1;
62             % Aggiornamento variabili in
63             matrice_nodi
64             matrice_nodi{w_nodo,3} = w_nodo-1;
65             matrice_nodi{w_nodo,1} =
66                 tab_generale_new{i,1};
67             matrice_nodi{w_nodo,2} =
68                 tab_generale_new{i,2};
69         end
70     end
71 end
72 end
73 end
74 end
75 % Inizializzazione variabile contatore 1
76 i = 0;
77 % Inizializzazione variabile contatore 2
78 j = 0;
```

```
68 % Inizializzazione contatore ciclo matrice_nodi
69     k = 0;
70 % Calcolo della lunghezza della tabella generale
71     N = length(tab_generale_new);
72 % Calcolo della lunghezza della matrice_nodi
73     M = length(matrice_nodi);
74     for i=1:N
75         % Confronto la colonna END con la matrice_nodi
76         for j=1:N
77             % Condizione che mi verifica elementi non ripetibili
78             % Prendo il primo elemento lo confronto con altri e se
              trovo
79             % coincidenza ignoro se e' nuovo lo aggiungo alla lista
              verificando
80             % di non immettere gli stessi valori gia' riscontrati
              precedentemente
81             % Inizializzazione verifica ad ogni iterazione
82             flag = 0;
83             % Calcolo della lunghezza della matrice_nodi
              variabile ogni ciclo
84             M = length(matrice_nodi);
85             if(isequal(i,j) == 0 && isempty(tab_generale_new{i
              ,3}) == 0 && isempty(tab_generale_new{i,4})== 0)
              %Non possono coincidere righe della stessa
              colonna e verifico se sono piene
86                 if(abs(tab_generale_new{i,3}-tab_generale_new{j
              ,3}) > epsilon)
87                     if(abs(tab_generale_new{i,4}-
              tab_generale_new{j,4}) > epsilon)
88                         % Verificato che la prima colonna ha
              valore univoco
89                         % differente prendo questo valore e se
              passa il
90                         % test di check della matrice colonne e'
              ok e lo
91                         % posso inserire (non ho copioni sulla
              matrice_nodi
92                         % flag = 0, altrimenti flag = 1)
93                         % Significa che la matrice_nodi e' piena
              e posso iniziare a confrontarla
94                         if(w_nodo>2)
```

```

95         % Test di check con tolleranza
          imposta
96         for k = 2:M
97             if(abs(tab_generale_new{i,3}-
                    matrice_nodi{k,1})< epsilon)
98                 if(abs(tab_generale_new{i
                    ,4}-matrice_nodi{k,2})<
                    epsilon)
99                     % Significa che ha trovato
                    un doppione e non
                    aggiorna in tabella
                    flag = 1;
100                 end
101             end
102         end
103     end
104 end
105 if(flag == 0)
106     % Incremento del contatore indice
        matrice_nodi
107     w_nodo = w_nodo+1;
108     % Aggiornamento variabili in
        matrice_nodi
109     matrice_nodi{w_nodo,3} = w_nodo-1;
110     matrice_nodi{w_nodo,1} =
        tab_generale_new{i,3};
111     matrice_nodi{w_nodo,2} =
        tab_generale_new{i,4};
112     end
113 end
114 end
115 end
116 end
117 end

```

12. Esamino le cassette ed escludo tutte le cabine prive di connessione con la rete esaminata:

```

1 % Estrapolazione del lato che va dal nodo virtuale (0,0) al nodo
  centro
2 % cassetta per ciascuna di essa

```

```
3 % Inizio il ciclo dalla riga successiva alla prima gia'  
   analizzata  
4 % Variabile ausiliaria contatore ciclo matrice  
   tab_layer_CASSETTE_new  
5   k = 0;  
6 % Variabile ausiliaria indice matrice tab_layer_CASSETTE_new  
   j = 1;  
7  
8 % Calcolo della lunghezza della tabella tab_layer_CASSETTE  
9   N = length(sequenza_layer_CASSETTE(:,1));  
10 % Inizializzazione ciclo di confronto notare che  
    tab_layer_CASSETTE  
11 % mantiene l'intestazione (giustificazione della riga 2)  
12 % Inizializzazione parametri di aggiornamento del ciclo notare  
    che  
13 % tab_layer_CASSETTE_new non mantiene l'intestazione  
14   varx_start = sequenza_layer_CASSETTE{2,1};  
15   vary_start = sequenza_layer_CASSETTE{2,2};  
16   varx_end = sequenza_layer_CASSETTE{2,3};  
17   vary_end = sequenza_layer_CASSETTE{2,4};  
18   var_R = sequenza_layer_CASSETTE{2,5};  
19   var_id = sequenza_layer_CASSETTE{2,6};  
20   tab_layer_CASSETTE_new{1,1} = sequenza_layer_CASSETTE{2,1};  
21   tab_layer_CASSETTE_new{1,2} = sequenza_layer_CASSETTE{2,2};  
22   tab_layer_CASSETTE_new{1,3} = sequenza_layer_CASSETTE{2,3};  
23   tab_layer_CASSETTE_new{1,4} = sequenza_layer_CASSETTE{2,4};  
24   tab_layer_CASSETTE_new{1,5} = sequenza_layer_CASSETTE{2,5};  
25   tab_layer_CASSETTE_new{1,6} = sequenza_layer_CASSETTE{2,6};  
26 % Stampa su tabella tab_layer_CASSETTE_new  
27   tab_layer_CASSETTE_new{j,1} = varx_start;  
28   tab_layer_CASSETTE_new{j,2} = vary_start;  
29   tab_layer_CASSETTE_new{j,3} = varx_end;  
30   tab_layer_CASSETTE_new{j,4} = vary_end;  
31   tab_layer_CASSETTE_new{j,5} = var_R;  
32   tab_layer_CASSETTE_new{j,6} = var_id;  
33   for k = 2:N  
34     varx_start_new = sequenza_layer_CASSETTE{k,1};  
35     vary_start_new = sequenza_layer_CASSETTE{k,2};  
36     varx_end_new = sequenza_layer_CASSETTE{k,3};  
37     vary_end_new = sequenza_layer_CASSETTE{k,4};  
38     var_R_new = sequenza_layer_CASSETTE{k,5};  
39     var_id_new = sequenza_layer_CASSETTE{k,6};
```

```

40 % Condizione tale per cui riscontra differente
    identificazione tra elementi in differenti cassette
41 if(strcmp(var_id_new,var_id)==0)
42 % Incremento variabile ausiliaria indice matrice
    tab_layer_CASSETTE_new
43     j = j+1;
44 % Aggiornamento Variabili
45     varx_start = sequenza_layer_CASSETTE{k,1};
46     vary_start= sequenza_layer_CASSETTE{k,2};
47     varx_end = sequenza_layer_CASSETTE{k,3};
48     vary_end = sequenza_layer_CASSETTE{k,4};
49     var_R = sequenza_layer_CASSETTE{k,5};
50     var_id = sequenza_layer_CASSETTE{k,6};
51 % Stampa su tabella tab_layer_CASSETTE_new
52     tab_layer_CASSETTE_new{j,1} = varx_start;
53     tab_layer_CASSETTE_new{j,2} = vary_start;
54     tab_layer_CASSETTE_new{j,3} = varx_end;
55     tab_layer_CASSETTE_new{j,4} = vary_end;
56     tab_layer_CASSETTE_new{j,5} = var_R;
57     tab_layer_CASSETTE_new{j,6} = var_id;
58 end
59 end
60 % tab_layer_CASSETTE_new esclude le cassette isolate senza
    connessioni
61 % dalla lista tab_layer_CASSETTE togliendone anche l'
    intestazione

```

13. Per ciascun lato di cassetta ricostruito associo un interruttore configurabile dal committente tramite apposita interfaccia:

```

1 % Variabile ausiliaria contatore ciclo matrice
    tab_layer_CAVI_POSITIVI
2     k = 0;
3 % Variabile ausiliaria indice matrice tab_layer_CASSETTE_new
4     j = 0;
5 % Variabile ausiliaria indice matrice tab_INTERRUTTORE
6     w = 1;
7     flag = 0;
8 % Fase di print degli interruttori se print_new = 1
9     if (print_new == 1)
10        for k =1:length(tab_layer_CAVI_POSITIVI(:,1))

```

```
11     for j=1:length(tab_layer_CASSETTE_new(:,1))
12     % Verifico che il cavo positivo in START rientri nella
13     % tolleranza
14     if(abs(tab_layer_CAVI_POSITIVI{k,1}-
15         tab_layer_CASSETTE_new{j,3})<tolleranza_cassetta)
16         if(abs(tab_layer_CAVI_POSITIVI{k,2}-
17             tab_layer_CASSETTE_new{j,4})<tolleranza_cassetta)
18             if (flag == 0)
19                 tab_EXP_INTERRUTTORE{w,1} = 1;
20                 name_INTERRUTTORE = strcat('INT-',
21                     tab_layer_CASSETTE_new{j,6});
22                 name_INTERRUTTORE = strcat(name_INTERRUTTORE
23                     , ' - ');
24                 name_INTERRUTTORE = strcat(name_INTERRUTTORE
25                     , tab_layer_CAVI_POSITIVI{k,6});
26                 tab_EXP_INTERRUTTORE{w,2} =
27                     name_INTERRUTTORE;
28                 w = w+1;
29                 flag = 1;
30             end
31         end
32     end
33     % Verifico che il cavo positivo in END rientri nella
34     % tolleranza
35     if(abs(tab_layer_CAVI_POSITIVI{k,3}-
36         tab_layer_CASSETTE_new{j,3})<tolleranza_cassetta)
37         if(abs(tab_layer_CAVI_POSITIVI{k,4}-
38             tab_layer_CASSETTE_new{j,4})<tolleranza_cassetta)
39             if (flag == 0)
40                 tab_EXP_INTERRUTTORE{w,1} = 1;
41                 name_INTERRUTTORE = strcat('INT-',
42                     tab_layer_CASSETTE_new{j,6});
43                 name_INTERRUTTORE = strcat(name_INTERRUTTORE
44                     , ' - ');
45                 name_INTERRUTTORE = strcat(name_INTERRUTTORE
46                     , tab_layer_CAVI_POSITIVI{k,6});
47                 tab_EXP_INTERRUTTORE{w,2} =
48                     name_INTERRUTTORE;
49                 w = w+1;
50                 flag = 1;
51             end
52         end
53     end
```

```
38         end
39     end
40     flag = 0;
41     end
42 end
43 fid = fopen('CABINE_INTERRUTTORI.csv','w');
44 for j = 1:length(tab_EXP_INTERRUTTORE(:,1))
45     fprintf(fid, '%d,%s \n',tab_EXP_INTERRUTTORE{j,1},
46             tab_EXP_INTERRUTTORE{j,2});
47 end
48 fclose(fid);
49 end
50 % Fase di modellizzazione degli interruttori se print_new = 0
51 % Variabile modificabile di resistenza di interruttore
52 % modificabile
53 % dall'utente
54 % tab_INTERRUTTORE conterra' per ogni cavo positivo connesso
55 % entro la tolleranza
56 % specificata di cassetta un ramo interruttore con resistenza
57 % dal valore di
58 %  $1 \cdot 10^{-6}$  [ohm] che implica sia attivo con resistenza elevata
59 % si puo'
60 % supporre circuito aperto.
61 % Variabile ausiliaria contatore ciclo matrice
62 % tab_layer_CAVI_POSITIVI
63 k = 0;
64 % Variabile ausiliaria indice matrice tab_layer_CASSETTE_new
65 j = 0;
66 % Variabile ausiliaria indice matrice tab_INTERRUTTORE
67 w = 1;
68 flag = 0;
69 % Variabile modificabile di resistenza di interruttore
70 % modificabile
71 % dall'utente
72 % Apertura del File .csv
73 fid = fopen('CABINE_INTERFACCIA_UTENTE.csv');
74 % Lettura e scansione in tabella
75 Colonne_layer_CABINE_INTERRUTTORI = textscan(fid, '%S%S', '
76     delimiter',';');
77 % Chiusura del File .csv
78 fclose(fid);
```

```

71 s = 1;
72 for j = 2:length(Colonne_layer_CABINE_INTERRUTTORI(:,1))
73     tab_R_INTERRUTTORE{j-1,1} = str2double(
74         Colonne_layer_CABINE_INTERRUTTORI{1,1}{j,1});
75     tab_R_INTERRUTTORE{j-1,2} =
76         Colonne_layer_CABINE_INTERRUTTORI{1,2}{j,1};
77     if(isequal(tab_R_INTERRUTTORE{j-1,1},1) == 1)
78         tab_R_INTERRUTTORE{j-1,1} = R_INTERRUTTORE_ON;
79     end
80     if(isequal(tab_R_INTERRUTTORE{j-1,1},0) == 1)
81         tab_R_INTERRUTTORE{j-1,1} = R_INTERRUTTORE_OFF;
82     end
83 end
84 for k =1:length(tab_layer_CAVI_POSITIVI(:,1))
85     for j=1:length(tab_layer_CASSETTE_new(:,1))
86         % Verifico che il cavo positivo in START rientri nella
87         % tolleranza
88         if(abs(tab_layer_CAVI_POSITIVI{k,1}-
89             tab_layer_CASSETTE_new{j,3})<tolleranza_cassetta)
90             if(abs(tab_layer_CAVI_POSITIVI{k,2}-
91                 tab_layer_CASSETTE_new{j,4})<tolleranza_cassetta)
92                 if (flag == 0)
93                     tab_INTERRUTTORE{w,1} =
94                         tab_layer_CASSETTE_new{j,3};
95                     tab_INTERRUTTORE{w,2} =
96                         tab_layer_CASSETTE_new{j,4};
97                     tab_INTERRUTTORE{w,3} =
98                         tab_layer_CAVI_POSITIVI{k,1};
99                     tab_INTERRUTTORE{w,4} =
100                         tab_layer_CAVI_POSITIVI{k,2};
101                     tab_INTERRUTTORE{w,5} = tab_R_INTERRUTTORE{s
102                         ,1};
103                     tab_INTERRUTTORE{w,6} = tab_R_INTERRUTTORE{s
104                         ,2};
105                     w = w+1;
106                     s = s+1;
107                     flag = 1;
108                 end
109             end
110         end
111     end
112 end

```

```

100 % Verifico che il cavo positivo in END rientri nella
101 tolleranza
102 if(abs(tab_layer_CAVI_POSITIVI{k,3}-
103 tab_layer_CASSETTE_new{j,3})<tolleranza_cassetta)
104 if(abs(tab_layer_CAVI_POSITIVI{k,4}-
105 tab_layer_CASSETTE_new{j,4})<tolleranza_cassetta)
106 if (flag == 0)
107     tab_INTERRUTTORE{w,1} =
108         tab_layer_CASSETTE_new{j,3};
109     tab_INTERRUTTORE{w,2} =
110         tab_layer_CASSETTE_new{j,4};
111     tab_INTERRUTTORE{w,3} =
112         tab_layer_CAVI_POSITIVI{k,3};
113     tab_INTERRUTTORE{w,4} =
114         tab_layer_CAVI_POSITIVI{k,4};
115     tab_INTERRUTTORE{w,5} = tab_R_INTERRUTTORE{s
116         ,1};
117     tab_INTERRUTTORE{w,6} = tab_R_INTERRUTTORE{s
118         ,2};
119     w = w+1;
120     s = s+1;
121     flag = 1;
122 end
123 end
124 end
125 flag = 0;
126 end
127 end

```

14. Effettuo la procedura di calcolo della matrice delle incidenze con le configurazioni di rete imposte:

```

1 % Tolgo l'intestazione a matrice_nodi
2 matrice_nodi(1,:) = [];
3 % Generazione matrice_lati
4 matrice_lati = [tab_layer_CASSETTE_new;
5 tab_layer_CAVI_POSITIVI; tab_layer_CAVI_ALIMENTAZIONE;
6 tab_layer_LA; tab_INTERRUTTORE; tab_layer_LAME;
7 tab_layer_TRAM(:,(1:6))];
8 % Generazione della matrice delle incidenze

```

```
6 % Confronto la matrice_nodi con la matrice_lati se ho
   coincidenza nello
7 % START avro' -1, mentre se ho coincidenza nell'END avro' +1, se
   infine non ho
8 % correlazione ne' con start, ne' con end avro' 0.
9 % Inizializzazione delle variabili ausiliarie
10 % Inizializzazione variabile ausiliaria contatore indice
11     k = 1;
12 % Inizializzazione variabile ausiliaria contatore nodi
13     i = 0;
14 % Inizializzazione variabile ausiliaria contatore lati
15     j = 0;
16 % Calcolo della lunghezza della matrice_nodi
17     N = length(matrice_nodi);
18 % Calcolo della lunghezza della matrice_lati
19     M = length(matrice_lati);
20 % Variabile di sistema numero di nodi
21     Num_nodi = N;
22 % Variabile di sistema numero di lati
23     Num_lati = M;
24     flag = 0;
25 % Sfrutto un approccio di puntatori piuttosto che matriciale per
   memorizzare
26 % la posizione relativa degli indici e risparmiare
   memorizzazione metodo
27 % efficace che usero' successivamente nella ricostruzione e'
   matrix sparse
28     for i=1:N
29         for j = 1:M
30             % Ricerca coincidenza nello START
31             if(abs(matrice_nodi{i,1}-matrice_lati{j,1})< epsilon)
32                 % Aggiornamento matrice_incidenze{i,j} = 1;
33                 if(abs(matrice_nodi{i,2}-matrice_lati{j,2})< epsilon
34                     )
35                     if(flag == 0)
36                         puntatore{k,1} = i;
37                         puntatore{k,2} = j;
38                         puntatore{k,3} = -1;
39                         k = k+1;
39                         % Incremento variabile ausiliaria contatore
                           indice
```

```

40         flag = 1;
41         end
42     end
43 end
44 % Ricerca coincidenza nell'END
45 if(abs(matrice_nodi{i,1}-matrice_lati{j,3})< epsilon)
46     % Aggiornamento matrice_incidenze{i,j} = -1;
47     if(abs(matrice_nodi{i,2}-matrice_lati{j,4})< epsilon
48         )
49         if(flag == 0)
50             puntatore{k,1} = i;
51             puntatore{k,2} = j;
52             puntatore{k,3} = 1;
53             k = k+1;
54             % Incremento variabile ausiliaria contatore
55             indice
56             flag = 1;
57         end
58     end
59 end
60 flag = 0;
61 end
62 % Ricostruisco la matrice delle incidenze dai puntatori per
63 invertirla
64 % Matrice delle incidenze matrice_incidenze
65 % Inizializzazione variabile ausiliaria contatore
66 matrice_incidenze
67 k = 0;
68 % Inizializzazione matrice_incidenze tipo sparse
69 j = 0;
70 matrice_incidenze = sparse(Num_nodi,Num_lati);
71 for k = 1:length(puntatore)
72     % Calcolo la lunghezza del vettore puntatore
73     if(puntatore{k,3}==1)
74         matrice_incidenze((puntatore{k,1}),(puntatore{k,2}))
75             = 1;
76     end
77     if(puntatore{k,3}== -1)
78         matrice_incidenze((puntatore{k,1}),(puntatore{k,2}))
79             = -1;

```

```

75     end
76     end
77 % Passo da una matrice di tipo sparse ad una normale matrice
    tipo double
78     matrice_incidenze = full(matrice_incidenze);
79 % Mi riconduco alla matrice delle incidenze ridotta
80 % Matrice delle incidenze ridotta matrice_incidenze_ridotta
81     matrice_incidenze_rid = matrice_incidenze;
82 % Decido di togliere il primo lato della cassetta che va dal nodo
    fittizio (0,0) al centro della cassetta
83     matrice_incidenze_rid(1,:) = [];

```

15. Effettuo il calcolo automatico della rete

```

1 % Creo il vettore delle resistenze
2 % Calcolo la lunghezza della matrice_lati
3     N = length(matrice_lati);
4 % Variabile ausiliaria contatore matrice_lati
5     k = 0;
6     for k = 1:N
7         % Calcolo il vettore delle conduttanze
8             vett_R(k,1) = matrice_lati{k,5};
9             vett_G(k,1) = 1/vett_R(k,1);
10        end
11 % Diagonalizzo il vettore delle conduttanze
12     G = diag(vett_G);
13 % Definisco una nomenclatura univoca per il calcolo automatico
    della
14 % corrente nella rete:
15 % 1) is, vettore dei generatori di corrente ai nodi
16 % 2) A, matrice delle incidenze ridotta
17 % 3) Y, matrice delle ammettenze ai nodi
18 % 4) vs, vettore dei generatori di tensione ai lati
19 % 5) js, vettore dei generatori di corrente ai lati
20 % 6) G, matrice delle ammettenze di lato
21 % 7) e, vettore delle tensioni ai nodi
22 % 8) v, vettore delle tensioni di lato
23 % 9) jlato, vettore delle correnti di lato
24 % Definizione nomenclatura matrice delle incidenze ridotta
25     A = matrice_incidenze_rid;
26 % Ricostruisco vs, vettore dei generatori di tensione ai lati

```

```
27 % Inizializzazione variabile ausiliaria tab_layer_CASSETTE_new
28     k = 0;
29     for k = 1: length(tab_layer_CASSETTE_new(:,1))
30         vs(k,1) = V_CASSETTE;
31     end
32     while(k<Num_lati)
33         k = k+1;
34         vs(k,1) = 0;
35     end
36 % Ricostruisco js, vettore dei generatori di corrente ai lati
37 % Inizializzazione variabile ausiliaria tab_layer_TRAM
38     k = 0;
39     for k = 1:Num_lati-length(tab_layer_TRAM(:,1))
40         js(k,1) = 0;
41     end
42     while(k<Num_lati)
43         k = k+1;
44         js(k,1) = I_TRAM;
45     end
46 % Procedura di calcolo automatico delle correnti della rete
47 % Calcolo is, vettore dei generatori di corrente ai nodi
48     is = A * G * vs - ( A * js );
49 % Calcolo Y, matrice delle ammettenze ai nodi
50 % Transpose consente di effettuare la trasposizione non
    complessa coniugata
51     Y = A * G * transpose(A);
52 % Calcolo e, vettore delle tensioni ai nodi
53 % Metodo alternativo efficace che richiede trasposizione
    rispetto alla funzione MATLAB inv()
54     e = pinv(Y)*is;
55 % Calcolo v, vettore delle tensioni di lato
56 % Transpose consente di effettuare la trasposizione non
    complessa coniugata
57     v = transpose(A) * e;
58 % Calcolo jlato, vettore delle correnti di lato e lo approssimo
59     jlato = round(G * v + js - G * vs);
60     v = round(v);
```

16. Realizzo una matrice generale che raccolga in maniera coerente un sunto di tutte le elaborazioni svolte in maniera coerente:

```

1  % Inizializzazione variabile contatore matrice_lati
2     i = 0;
3  % Inizializzazione variabile contatore matrice_nodi
4     j = 0;
5  % Inizializzazione contatore indice matrice_generale START
6     k = 2;
7  % Inizializzazione contatore indice matrice_generale END
8     w = 2;
9  % Intestazione matrice_generale
10    matrice_generale{1,1} = 'NUM - NODO PARTENZA';
11    matrice_generale{1,2} = 'NUM - NODO ARRIVO';
12    matrice_generale{1,3} = 'CORRENTE DI LATO [A]';
13    matrice_generale{1,4} = 'POTENZA MEDIA LATO [kW]';
14    matrice_generale{1,5} = 'PERDITE LATO [W]';
15    matrice_generale{1,6} = 'ID - LATO';
16    matrice_generale{1,7} = 'X - NODO PARTENZA';
17    matrice_generale{1,8} = 'Y - NODO PARTENZA';
18    matrice_generale{1,9} = 'X - NODO ARRIVO';
19    matrice_generale{1,10} = 'Y - NODO ARRIVO';
20    for i = 1:length(matrice_lati(:,1))
21        matrice_generale{i+1,7} = matrice_lati{i,1};
22        matrice_generale{i+1,8} = matrice_lati{i,2};
23        matrice_generale{i+1,9} = matrice_lati{i,3};
24        matrice_generale{i+1,10} = matrice_lati{i,4};
25    for j = 1:Num_nodi
26        % Coincidenza con START
27        if(abs(matrice_lati{i,1}-matrice_nodi{j,1})< epsilon
28            )
29            if(abs(matrice_lati{i,2}-matrice_nodi{j,2})<
30                epsilon)
31                matrice_generale{i+1,1} = matrice_nodi{j,3};
32            end
33        end
34        % Coincidenza con END
35        if(abs(matrice_lati{i,3}-matrice_nodi{j,1})< epsilon
36            )
37            if(abs(matrice_lati{i,4}-matrice_nodi{j,2})<
38                epsilon)

```

```

35         matrice_generale{i+1,2} = matrice_nodi{j,3};
36         end
37     end
38 end
39 % Aggiornamento variabili in tabella
40 matrice_generale{i+1,3} = jlato(i,1);
41 matrice_generale{i+1,6} = matrice_lati{i,6};
42 end
43 % Funzione per il calcolo della potenza media di lato come la
44 % la tensione ai 2 estremi dei nodi e per il calcolo delle
45 % perdite.
46 flag = 0;
47 for j = 1:length(matrice_lati(:,1))
48     if(isequal(matrice_generale{j+1,1},1)==1)
49         matrice_generale{j+1,4} = abs(round((v(j,1)*abs(
50             matrice_generale{j+1,3}))*10^(-3)));
51         matrice_generale{j+1,5} = 0;
52         flag = 1;
53     end
54     if(flag == 0)
55         matrice_generale{j+1,4} = abs(round((((e(
56             matrice_generale{j+1,1}-1))*matrice_generale{j+1,3}) +
57             ((e(matrice_generale{j+1,2}-1))*abs(matrice_generale{
58                 j+1,3}))/2)*10^(-3)));
59         matrice_generale{j+1,5} = abs(round(((e(matrice_generale{
60             j+1,1}-1) - e(matrice_generale{j+1,2}-1))*abs(
61             matrice_generale{j+1,3})));
62     end
63     flag = 0;
64 end

```

17. Per la visualizzazione e l'esportazione dei dati in QGIS effettuo delle verifiche di coerenza dei risultati come step finale del codice. Fase finale del software è la verifica che il tram sia effettivamente connesso alla rete nei calcoli e la stampa coerente dei dati per l'esportazione e successiva simulazione in ambiente QGIS

```

1 % Verifica se i risultati sono affidabili
2     if(flag_noTRAM == 1)
3         clc;
4         messaggio = ('Attenzione TRAM non rilevato')

```

```
5     end
6 % Esportazione del file .csv per importazione in QGIS
7     if(flag_noTRAM == 0)
8         % Conservazione intestazione tabulare per stampa
9             intestazione = matrice_generale(1,:);
10            matrice_generale(1,:) = [];
11        % Eliminazione intestazione tabulare per esportazione
12            flag = 0;
13        % Variabile ausiliaria cavi positivi
14            i = 1;
15        % Variabile ausiliaria cavi alimentazione
16            k = 1;
17        % Variabile ausiliaria linea aerea
18            w = 1;
19        % Variabile ausiliaria interruttore
20            p = 1;
21        % Variabile ausiliaria lame
22            q = 1;
23            fid = fopen('layer_INTERRUTTORI_LAME.csv','w');
24            Num_Line = length(tab_layer_CASSETTE_new(:,1));
25            Num_Line1 = Num_Line + length(tab_layer_CAVI_POSITIVI
26                (:,1));
27            Num_Line2 = Num_Line1 + length(
28                tab_layer_CAVI_ALIMENTAZIONE(:,1));
29            Num_Line3 = Num_Line2 + length(tab_layer_LA(:,1));
30            Num_Line4 = Num_Line3 + length(tab_INTERRUTTORE(:,1));
31            for j = 1:length(matrice_generale(:,1))
32                % Cabina
33                % Tra 0 e Num_Line
34                if(j<=Num_Line)
35                    flag = 1;
36                end
37                % Cavi Positivi
38                % Tra Num_Line+1 e Num_Line1
39                if(j>Num_Line && j<=Num_Line1)
40                    flag = 2;
41                end
42                % Cavi di Alimentazione
43                % Tra Num_Line1+1 e Num_Line2
44                if(j>Num_Line1 && j<=Num_Line2)
45                    flag = 3;
```

```
44         end
45         % Linea Aerea
46         % Tra Num_Line2+1 e Num_Line3
47         if(j>Num_Line2 && j<=Num_Line3)
48             flag = 4;
49         end
50         % Interruttori
51         % Tra Num_Line3+1 e Num_Line4
52         if(j>Num_Line3 && j<=Num_Line4)
53             flag = 5;
54         end
55         % Si aggiungono interruttori a ciascuna cassetta e lame
56         % oltre al tram ma escludo il nodo virtuale in
57         % riferimento al centro cassetta e al tram quindi li
58         % specifico
59         % flag = 1 e pongo la condizione di verifica cautelativa
60         % Lame e Tram
61         if(j>Num_Line4)
62             flag = 6;
63         end
64         % Escludo nell'esportazione i riferimenti di tram e
65         % cassetta al
66         % potenziale virtuale
67         if(isequal(matrice_generale{j,1},1)==1)
68             flag = 1;
69         end
70         if(isequal(matrice_generale{j,2},1)==1)
71             flag = 1;
72         end
73         % Scrittura SHAPEFILE CAVI POSITIVI
74         if(flag == 2)
75             [A_CPOS.Geometry] = S_CPOS.Geometry;
76             [A_CPOS.X] = S_CPOS.X;
77             [A_CPOS.Y] = S_CPOS.Y;
78             A_CPOS(i).Correnti = matrice_generale{j,3};
79             A_CPOS(i).Potenza = matrice_generale{j,4};
80             A_CPOS(i).Perdite = matrice_generale{j,5};
81             A_CPOS(i).Attributi = matrice_generale{j,6};
82             shapewrite(A_CPOS(:), 'CAVI_POSITIVI_linee.shp');
83             i = i+1;
84         end
```

```
83 % Scrittura SHAPEFILE CAVI ALIMENTAZIONE
84 if(flag == 3)
85 [A_CALIM.Geometry] = S_CALIM.Geometry;
86 [A_CALIM.X] = S_CALIM.X;
87 [A_CALIM.Y] = S_CALIM.Y;
88 A_CALIM(k).Correnti = matrice_generale{j,3};
89 A_CALIM(k).Potenza = matrice_generale{j,4};
90 A_CALIM(k).Perdite = matrice_generale{j,5};
91 A_CALIM(k).Attributi = matrice_generale{j,6};
92 shapewrite(A_CALIM(:), 'CAVI_ALIMENTAZIONE_linee.shp');
93 k = k+1;
94 end
95 %Scrittura SHAPEFILE LINEA AEREA
96 if(flag == 4)
97 [A_LA.Geometry] = S_LA.Geometry;
98 [A_LA.X] = S_LA.X;
99 [A_LA.Y] = S_LA.Y;
100 A_LA(w).Correnti = matrice_generale{j,3};
101 A_LA(w).Potenza = matrice_generale{j,4};
102 A_LA(w).Perdite = matrice_generale{j,5};
103 A_LA(w).Attributi = matrice_generale{j,6};
104 shapewrite(A_LA(:), 'LA_GENERALE_linee.shp');
105 w = w+1;
106 end
107 if(flag == 5)
108 % Scrivo uno SHAPEFILE esportabile per relativi
109 % INTERRUTTORI
110 [S_INT(p).Geometry] = 'Polyline';
111 [S_INT(p).X(1)] = matrice_generale{j,7};
112 [S_INT(p).Y(1)] = matrice_generale{j,8};
113 [S_INT(p).X(2)] = matrice_generale{j,9};
114 [S_INT(p).Y(2)] = matrice_generale{j,10};
115 S_INT(p).Correnti = matrice_generale{j,3};
116 S_INT(p).Potenza = matrice_generale{j,4};
117 S_INT(p).Perdite = matrice_generale{j,5};
118 S_INT(p).Attributi = matrice_generale{j,6};
119 shapewrite(S_INT(:), 'layer_INTERRUTTORI.shp');
120 p = p+1;
121 end
122 if(flag == 6)
123 % Scrivo uno SHAPEFILE esportabile per relative LAME
```

```
123     [S_LM(q).Geometry] = 'Polyline';
124     [S_LM(q).X(1)] = matrice_generale{j,7};
125     [S_LM(q).Y(1)] = matrice_generale{j,8};
126     [S_LM(q).X(2)] = matrice_generale{j,9};
127     [S_LM(q).Y(2)] = matrice_generale{j,10};
128     S_LM(q).Correnti = matrice_generale{j,3};
129     S_LM(q).Potenza = matrice_generale{j,4};
130     S_LM(q).Perdite = matrice_generale{j,5};
131     S_LM(q).Attributi = matrice_generale{j,6};
132     shapewrite(S_LM(:), 'layer_LAME.shp');
133     q = q+1;
134     end
135     flag = 0;
136 end
137 fclose(fid);
138 % Esportazione dei nodi su QGIS sottoforma di SHAPEFILE
139 for j = 1:length(e(:,1))
140     [S_NODI(j).Geometry] = 'Point';
141     [S_NODI(j).X(1)] = matrice_nodi{j+1,1};
142     [S_NODI(j).Y(1)] = matrice_nodi{j+1,2};
143     [S_NODI(j).Tensioni(1)] = abs(round(e(j,1)));
144     [S_NODI(j).Nome(1)] = matrice_nodi{j+1,3};
145     shapewrite(S_NODI(:), 'layer_NODI.shp');
146 end
147 % Reinserimento di intestazione tabulare
148 matrice_generale = [intestazione; matrice_generale];
149 clc;
150 messaggio = ('Calcolo effettuato con successo')
151 % Print sulla command window per una lettura dei dati di
    interesse
152     matrice_generale
153 % Salvataggio delle variabili di simulazione
154     save('risultati_simulazione');
155 end
```

Appendice B

Licenze Materiale Multimediale

Si citano le licenze per gli elementi multimediali inseriti all'interno della tesi:
QGIS - Plugin utilizzato per il progetto.
Comune di Torino - Geoportale Dati - Creative Commons public licence Attribuzione - 2.5 (ITALIA).
Immagine Ale Sasso - Creative Commons Attribution - Share Alike 3.0 Unported.
Immagine Luigi Chiesa - Creative Commons Attribution - Share Alike 3.0 Unported.
Immagine Nicolò Lazzati - Creative Commons Attribution 2.0 Generic (CC BY 2.0).
Tutto il materiale pubblicato ed il logo Augusta Tauringrid è sottoposto a licenza Creative Commons Attribuzione 4.0 Internazionale (CC BY 4.0) to Bertolone Citin Lorenzo.

Ringraziamenti

*Si ringraziano i miei relatori **Prof. E.Pons** e **Ing. P.Colella** che mi hanno seguito ed aiutato nella stesura e rielaborazione della mia tesi di laurea. La Loro disponibilità, pazienza e continua e perseverante cortesia mi ha accompagnato per tutto il progetto e rivolgo un caloroso ringraziamento per tutto il tempo che mi hanno dedicato. Ringraziamenti particolari all' **Ing. R.Rizzoli** e all'ufficio **Infra.To** per il tempo e la disponibilità dedicata a me durante la realizzazione del progetto.*