

Healing Touch for Power Systems (HITS)

Version 1.0

May 9, 2018

Zhijun Qin, Yunhe Hou

The University of Hong Kong

All Rights Reserved

SOFTWARE DESCRIPTION

The Healing Touch for System (HITS) version 1.0 is a MATLAB toolbox that can establish a system restoration plan following a complete or partial outage.

Description

The HITS is a MATLAB toolbox that can establish a system restoration plan following a complete or partial outage. Users can configure the characteristics of generating units, loads, transmission lines, and transformers. The restoration process can be established automatically. The users can also control the restoration process by setting milestone and priority of each generating unit and load. The plan can be displayed in MATLAB Command Window, and all the results are stored in a structural variable named "Grid".

Benefits and Value

Value and benefits provided by the Generic Restoration Milestones are:

- Establishing of a system restoration planning from complete or partial outage.
- Evaluating a system restoration plan and checking constraints during restoration process.
- Displaying a system restoration plan in MATLAB Command Window.

System Requirements

HITS 1.0 can run on Windows XP, Windows Vista and Windows 7 (32-Bit and 64-Bit) and on MATLAB 7.0 or higher.

Citing HITS

While not required by the terms of the license, we do request that publications derived from the use of HITS explicitly acknowledge that fact by citing the following reference

Y. Hou, C.-C. Liu, K. Sun, P. Zhang, S. Liu, and D. Mizumura, "Computation of Milestones for Decision Support during System Restoration," *Power Systems, IEEE Transactions on*, vol. 26, pp. 1399-1409, 2011

License and Terms of Use

The code in HITS is distributed under the 3-clause BSD license¹. The full text of the license can be found at the top level of the distribution.

HITS is distributed in the hope that it will be useful, but WITHOUT ANY WARRANTY.

¹ <https://opensource.org/licenses/BSD-3-Clause>

ABSTRACT

The developed HITS program is a MATLAB toolbox that can establish a system restoration plan following a complete or partial outage. The plan can be displayed in MATLAB Command Window, and all the results are stored in a structural variable named “Grid”. Users can also control the restoration process by setting milestone and priority for each generating unit and load.

This user’s manual provides:

- Instructions for installing and running HITS,
- Detailed explanation for the operation of the program.

It is assumed that the user is familiar with Microsoft (MS) Windows operations and MATLAB environment. The first section provides an introduction to HITS. Section 2 gives the installation instruction for the toolbox. Section 3 describes how to use HITS.

Keywords

Power System Restoration

1 INTRODUCTION TO HITS

Power system restoration is well recognized as one of the most important tasks for electric power grids. Following a power outage, dispatchers in the control center work with the field crews to re-establish the generation and transmission systems and then to pick up load and restore service. It is reported that the impact of a blackout increases with the duration of its restoration. This program utilizes “generic restoration milestones” and is able to establish a system restoration plan following a complete or partial outage. This power system restoration plan can be established automatically and interactively. The following data (*.txt) are required:

- Topology and parameters of a power grid
- Initial state before system restoration
- Characteristics of each generating unit
- Characteristics of each load
- Characteristics of each branch
- Characteristics of each transformer

The program has the following functional features:

- The program establishes a system restoration plan following a complete or partial outage.
- Editable file (*.txt) allows the user to create new data files, enter or modify the input parameters.
- Calculation results are displayed in MATLAB Command Window and stored in a structural variable named “Grid”.

2 BEFORE USING HITS

This section describes the computer system requirements and installation procedures needed to use the HITS program.

2.1 Minimum Computer Requirements

A PC with the following system requirements:

- Windows XP, Windows Vista and Windows 7 (32-Bit and 64-Bit)
- 128 Megabytes RAM
- Monitor Resolution: 1024×768

MATLAB version requirements:

- MATLAB 7.0 or higher

2.2 Installation of HITS

The HITS is installed by copying the “HITS” folder (with its contents) to your system hard driver (e.g., C:).

2.3 Uninstall HITS program

To uninstall the HITS program:

Delete folder “HITS” from your hard driver.

3 OVERVIEW: THE HITS USERS INTERFACE

3.1 Directory Structure

After users have installed the HITS successfully, they will find five folders and five *.m files under the setup directory as shown in Figure 3-1.

- GridData: contains the power system grid data files and their parsing files.
- GRM1: contains the key functions of HITS.
- Initialization: contains the HITS variables initialization file and launch information file.
- OPF: contains the optimal power flow files.
- StepValue: update the variables step by step.
- GRMs_FileNameDef.m: defines data files' names and function files' names.
- GRMs_Main.m: the main algorithm file.

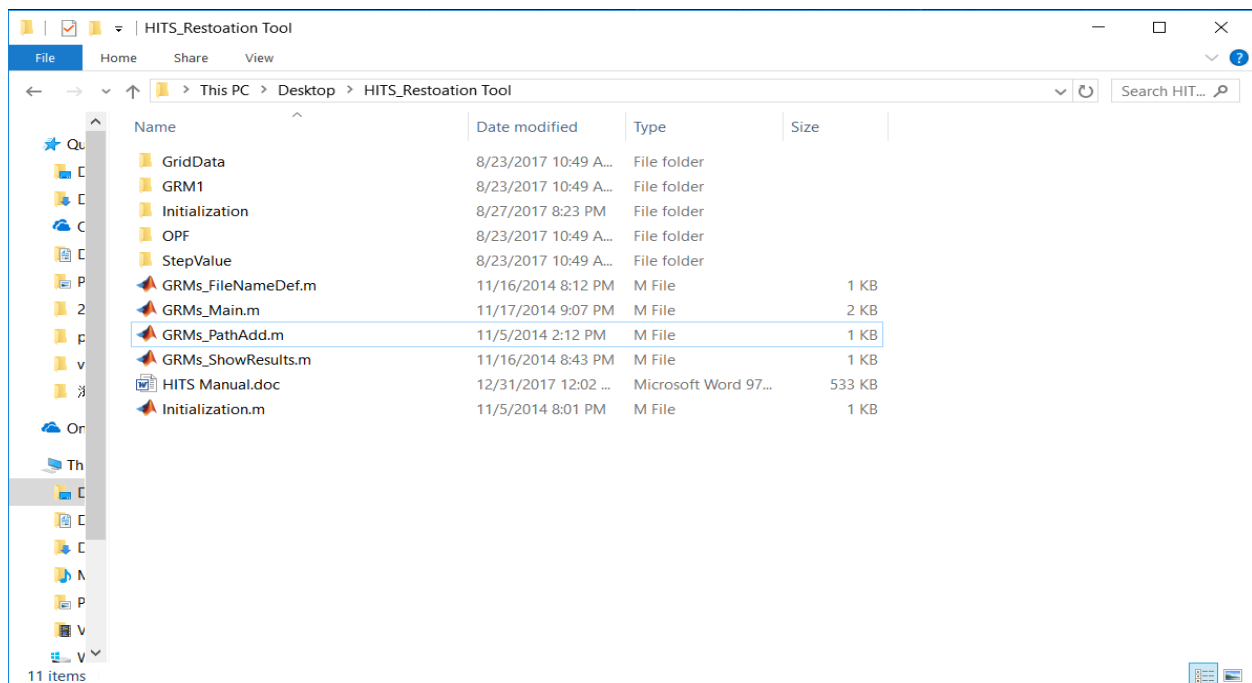


Figure 3-1
Program Files

3.2 Sample Data

The sample data files:

- RTS24_area.txt
- RTS24_bus.txt
- RTS24_criticalLoad.txt
- RTS24_dispatchableLoad.txt
- RTS24_gen.txt
- RTS24_line.txt
- RTS24_tran.txt

can be found under the “...\GRMs\GridData\RTS24” directory . The following chapter will show how to use the HITS program with the sample data.

3.3 Launching HITS

“GRMs_Main.m” is the main file of HITS, users can start HITS program by double clicking “GRMs_Main.m” in folder ,then clicking “Run” in MATLAB EDITOR view. Or the users can launch the MATLAB first, set the MATLAB current path to HITS setup directory, input *GRMs_Main* in the command window, then press “Enter” to start HITS.

Once users launch the program, the HITS information is displayed in command window, and then the calculation starts immediately with the sample data as shown in Figure 3-2.

All the details of grid restoration can be viewed in the structure variable *Grid*.

```
*****
                                <HITS>
                                Version 1.0
                                AUGUST 2017

    Copyright (c) 2017-2024, The University of Hong Kong.
    Copying and distribution of this file, with or without modif-
    ication, are permitted in any medium without royalty provide-
    d the copyright notice and this notice are preserved. This f-
    ile is offered as-is, without any warranty.
Host:      Matlab 8.5.0.197613 (R2015a)
Session:   07-Mar-2018 11:04:17
Path:      C:\Users\yunhe\Desktop\HITS_Restoation Tool
Author:    Yunhe Hou, Zhijun Qin, Chenchen Qian
Email:     yhhou@eee.hku.hk
Wbesite:   https://www.eee.hku.hk/~yhhou/
*****
```

Figure 3-2
HITS Launching View

Users can change the parameters, create new data files, and even modify the program codes by themselves. The following chapters will show how the HITS works, how the data defines.

4 CREATING CUSTOM DATA

4.1 Data Location

As with the Sample Data, Custom Data is also placed under the path: “...\GRMs\ GridData” as shown in Figure 4-1.

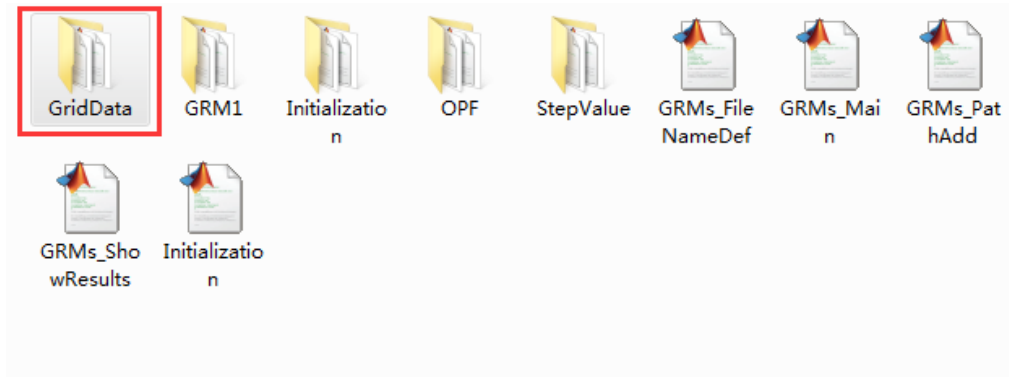


Figure 4-1
Data Location View

Users can create a new folder for storing new data files, as shown in Figure 4-2.

GRMData	2015/11/17 21:07	文件夹
MyData_1	2016/12/9 9:01	文件夹
RTS24	2015/11/17 21:06	文件夹
GRMs_GetData_Area	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_Bus	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_CriticalLoad	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_DispatchableLoad	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_Generator	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_Line	2015/11/17 23:45	MATLAB M-file
GRMs_GetData_Transformer	2015/11/17 23:46	MATLAB M-file
GRMs_Preprocessing	2014/11/5 20:42	MATLAB M-file

Figure 4-2
Create New Folder

4.2 Introduction of Data Format

In order to successfully establish the restoration process, users need to create the *Area*, *Bus*, *Critical Load*, *Dispatchable Load*, *Generator*, *Transformer*, and *Line* files. Each data file has a corresponding data format, just as follows.

- **Area Data File Format**

The first line is a comment string that starts with "%", and then the area data follows.

```
%      ID,      Name,      Buses
      1,      Area01, 24
```

ID (int) : Area ID;

Name (string) : Area Name;

Buses (int): Total Number of Buses in Current Area.

- **Bus Data File Format**

The first line is a comment string that starts with "%", and then the bus data follows.

```
%      ID,      Name,      Area,      V,      Delta
      1,      bus001,      1,      1,      0
      2,      bus002,      1,      1,      0
      3,      bus003,      1,      1,      0
      4,      bus004,      1,      1,      0
      5,      bus005,      1,      1,      0
      6,      bus006,      1,      1,      0
      7,      bus007,      1,      1,      0
      8,      bus008,      1,      1,      0
      .      .      .      .      .
```

ID (int) : Bus ID;

Name (string) : Bus Name;

Area (int): Which Area the Current Bus Belongs;

V(float): Initial Voltage Amplitude;

Delta(float): Initial Voltage Angle.

- **Critical Load Data File Format**

The first line is a comment string that starts with "%", and then the Critical Load data follows.

```
%      ID,      Name,      BusID,      P,      Q,      Priority
      1,      1,      19,      0.15,      0.08,      1
      2,      2,      14,      0.53,      0.19,      1
      3,      3,      9,      0.40,      0.1,      1
```

ID (int) : Critical Load ID;

Name (string) : Critical Load Name;

BusID (int): Which Bus the Current Critical Load Connected;

P(float): Active Power of Current Critical Load;

Q(float): Reactive Power of Current Critical Load;

Priority(int): Priority of Current Critical Load.

- **Dispatchable Load Data File Format**

The first line is a comment string that starts with "%", and then the dispatchable Load data follows.

%	ID,	Name,	BusID,	P,	Pmin,	Pmax,	Q,	Qmin,	Qmax,	Blocks
	1,	1,	17,	1.53,	0,	1.53,	0.68,	0,	0.68,	2
	2,	2,	11,	1.3725,	0,	1.3725,	0.64,	0,	0.64,	3
	3,	3,	4,	1.0,	0,	1.0,	0.3,	0,	0.3,	3
	4,	4,	10,	1.5,	0,	1.5,	0.5,	0,	0.5,	3

ID (int) : Dispatchable Load ID;

Name (string) : Dispatchable Load Name;

BusID (int): Which Bus the Current Dispatchable Load Connected;

P(float): Active Power of Current Dispatchable Load;

Pmin(float): Minim Active Power;

Pmax(float): Maximum Active Power;

Q(float): Reactive Power of Current Dispatchable Load;

Qmin(float): Minim Reactive Power;

Qmax(float): Maximum Reactive Power;

Blocks(int): The Number of Current Dispatchable Load can be divided;

- **Generator Data File Format**

The first line is a comment string that starts with "%", and then the generator data follows.

ID,	Name,	BusID,	Capacity,	BlackStartFlag,	RampingRate,	StartRequirement,	TimeToCrank(min),	Pmin,	Pmax,	MinOutputRate,	Priority
1,	1,	22,	3.85,	1,	3.84,	0,	0,	0,	999,	0.001,	1
2,	2,	21,	3.00,	0,	1.20,	0.066,	30,	0,	999,	0.1,	1
3,	3,	18,	6.60,	0,	3.46,	0.132,	30,	0,	999,	0.3,	1
4,	4,	16,	6.00,	0,	1.57,	0.12,	100,	0,	999,	0.5,	1
5,	5,	15,	2.52,	0,	1.50,	0.30,	120,	0,	999,	0.5,	1
6,	6,	13,	1.35,	0,	0.30,	0.027,	160,	0,	999,	0.3,	1
7,	7,	23,	3.00,	0,	1.20,	0.06,	100,	0,	200,	0.7,	1
8,	8,	7,	3.00,	0,	1.00,	0.09,	120,	0,	210,	0.5,	1
9,	9,	2,	3.45,	0,	1.48,	0.12,	30,	0,	240,	0.3,	1
10,	10,	1,	3.02,	0,	1.20,	0,	0,	0,	999,	0.1,	1

ID (int) : Generator ID;

Name (string) : Generator Name;

BusID (int): Which Bus the Generator Connected;

Capacity(float): Capacity of Current Generator;

BlackStartFlag(int): Black Start Flag of Current Generator;

RampingRate(float): Ramping Rate of Current Generator;

StartRequirement(float): Active Power needed When the Current Generator Start Working;

TimeToCrank(float): The Time from Generator Start Working to Start Cranking;

Pmin(float): Minim Active Power;

Pmax(float): Maximum Active Power;

MinOutputRate(float): Minim Output Rate .

- **Line Data File Format**

The first line is a comment string that starts with "%", and then the line data follows.

%	ID,	Name,	UseableFlag,	I,	J,	R,	X,	B,	Time
	1,	1,	1,	1,	2,	0.003,	0.014,	0.061,	5
	2,	2,	1,	1,	3,	0.055,	0.211,	0.057,	5
	3,	3,	1,	1,	5,	0.022,	0.085,	0.023,	5
	4,	4,	1,	2,	4,	0.033,	0.127,	0.034,	5
	5,	5,	1,	2,	6,	0.05,	0.192,	0.052,	5
	6,	6,	1,	3,	9,	0.031,	0.119,	0.032,	5
	7,	7,	1,	4,	9,	0.027,	0.104,	0.028,	5
	8,	8,	1,	5,	10,	0.023,	0.088,	0.024,	5
	9,	9,	1,	6,	10,	0.014,	0.061,	2.459,	5
	10,	10,	1,	7,	8,	0.016,	0.061,	0.017,	5
	11,	11,	1,	8,	9,	0.043,	0.165,	0.045,	5
	12,	12,	1,	8,	10,	0.043,	0.165,	0.045,	5
	13,	13,	1,	11,	13,	0.006,	0.048,	0.100,	5
	14,	14,	1,	11,	14,	0.005,	0.042,	0.088,	5
	15,	15,	1,	12,	13,	0.006,	0.048,	0.100,	5
	16,	16,	1,	12,	23,	0.012,	0.097,	0.203,	5
	17,	17,	1,	13,	23,	0.011,	0.087,	0.182,	5
	18,	18,	1,	14,	16,	0.005,	0.059,	0.082,	5
	19,	19,	1,	15,	16,	0.002,	0.017,	0.136,	5
	20,	20,	1,	15,	21,	0.003,	0.024,	0.206,	5
	21,	21,	1,	15,	24,	0.007,	0.052,	0.109,	5

ID (int) : Line ID;

Name (string) : Line Name;

UseableFlag (int): Initialization state of Line;

I(int): Left Bus Number of Line;

J(int): Right Bus Number of Line;

R(float): Line Resistance;

X(float): Line Reactance;

B(float): Line Acceptance;

Time(float): Line charging time.

- **Transformer Data File Format**

The first line is a comment string that starts with "%", and then the transformer data follows.

%	ID,	Name,	UseableFlag,	I,	J,	R,	X,	B,	Time
	1,	1,	1,	1,	2,	0.003,	0.014,	0.061,	5
	2,	2,	1,	1,	3,	0.055,	0.211,	0.057,	5
	3,	3,	1,	1,	5,	0.022,	0.085,	0.023,	5
	4,	4,	1,	2,	4,	0.033,	0.127,	0.034,	5
	5,	5,	1,	2,	6,	0.05,	0.192,	0.052,	5
	6,	6,	1,	3,	9,	0.031,	0.119,	0.032,	5
	7,	7,	1,	4,	9,	0.027,	0.104,	0.028,	5
	8,	8,	1,	5,	10,	0.023,	0.088,	0.024,	5
	9,	9,	1,	6,	10,	0.014,	0.061,	2.459,	5
	10,	10,	1,	7,	8,	0.016,	0.061,	0.017,	5
	11,	11,	1,	8,	9,	0.043,	0.165,	0.045,	5
	12,	12,	1,	8,	10,	0.043,	0.165,	0.045,	5
	13,	13,	1,	11,	13,	0.006,	0.048,	0.100,	5
	14,	14,	1,	11,	14,	0.005,	0.042,	0.088,	5
	15,	15,	1,	12,	13,	0.006,	0.048,	0.100,	5
	16,	16,	1,	12,	23,	0.012,	0.097,	0.203,	5
	17,	17,	1,	13,	23,	0.011,	0.087,	0.182,	5
	18,	18,	1,	14,	16,	0.005,	0.059,	0.082,	5
	19,	19,	1,	15,	16,	0.002,	0.017,	0.136,	5
	20,	20,	1,	15,	21,	0.003,	0.024,	0.206,	5
	21,	21,	1,	15,	24,	0.007,	0.052,	0.109,	5

ID (int) : Transformer ID;

Name (string) : Transformer Name;

UseableFlag (int): Initialization state of Transformer;

I(int): Left Bus Number of Transformer;

J(int): Right Bus Number of Transformer;

R(float): Transformer Resistance;

X(float): Transformer Reactance;

B(float): Transformer Acceptance;

Time(float): Transformer charging time.

5 INTRODUCTION OF KEY FUNCTIONS

- GRMs_Main.m: The main file.
- GRMs_FileNameDef.m: Used for file name definitions, including the data file name and the file name of some of the sub-function files.
- GRMs_PathAdd.m: Used to automatically add a path without having to add subdirectories manually.
- GRM1_LoopInit.m: As the initialization file of the main loop before GRM1 calculation, the variables which are the key to the loop are initialized.
- GRM1_StepInit.m: After a successful start of a generator, ready to start the next unit to do the initialization. Once a generating unit start successfully, step increase in the value of 1.
- GRM1_UpdateDepth.m: Update the search-depth D. In the current version, if all the generator sets cannot start at the current search-depth, then the step value continue the same while the search-depth D is enlarged, and the starting search of the generator sets is continued. If search-depth D has been expanded to the entire grid, and this stage can not successfully start the generator, then that the entire black start failed.
- GRM1_WaitForStart.m: Find the unit to be started under the current search dimension, schedule the load and the critical load.
- GRM1_SortGenerators.m: Prioritize the crews to be started.
- GRM1_CrankInit.m: Initialization before attempting to start a generator.
- GRM1_EnergizePath.m: The uncharged lines can be charged according to the generators, loads, and energized buses found in the search range, and multiple lines can be charged at the same time.
- GRM1_OPF.m: The main function of OPF.
- GRM1_CrankJudgment.m: Determine whether the current generator can be successfully started, and update some of the loop variables.
- GRM1_CalculateTime.m: Calculate the returned generator power based on the OPF, calculate the time of each generator, and return the total time.

6 SOFTWARE FLOW CHART

