CHAPTER 4

FAULT ZONE DETECTION

4.1 Introduction

The importance of devising the relays settings for individual fault locations has been emphasized in the literature as it yields faster relays operating time compared to the settings which are devised considering all possible fault locations [20] [21]. To implement this approach, the detection of fault occurrence by the protective relays must be followed by the detection of faulted zone (FZD) on which a fault has occurred. There are different FZD methods in the literature, where the methods based on sharing information between the end-relays via communication links are one of the popular methods. Different types of information such as differential current [93], phase angle [61], and directions of fault currents [20], [60] have been used to detect the fault. The major advantage of these methods is that they are independent of the fault current level, and thus will be suitable for both grid-connected and islanding mode. In [61], to detect the faulted zone, an additional section agent is proposed to install at the middle of each of the feeder zone. This section agent receives the phase angles from both end-relays for detecting the faulted zone. However, the raise in the number of section agents with the spreading of a network may put a question on the overall protection cost. In [20], each local downstream relay sends its fault current’s direction to the immediate previous upstream relay to detect the faulted zone. In [60], the end-relays send their fault current’s directions to the central unit to discover the faulted section. All these former approaches including [20] and [60] are fully dependent on the communication between the agents. The problem with these existing methods may occur if an end-relay or its communication link fails to function or send fault current's direction or phase-angle to either of the associated section agents, another end-relay or the central unit. In this situation, these FZD methods may fail to detect the faulted zone in the presence of incomplete information. Besides this, if any information is absent due to the detection of bad-data coming from the relays, these FZD methods may fail. In this thesis, these events of failing an end-relay and/or its communication link are named as FRC events. This chapter presents a new FZD method (named as DFZD method) for detecting the faulted zone in the presence of one or more than one FRC events or information loss. In this method, the relays are assumed to be equipped with advanced feature of sensing the direction of the fault current.

4.2 Proposed Problem Formulation

In this work, two bits, namely, Fixed Direction bit (FDb) and Relay Direction bit (RDb), have been generated to propose the problem formulation and solution method. A relay's FDb bit denotes a relay's reference direction which points towards the mid of the z^{th} zone. And, a relay's RDb bit denotes the direction of the fault current with respect to the corresponding FDb bit's direction. This section is divided into two sub-sections. The first sub-section describes the generating method of these bits. While the second sub-section demonstrates the basic existing FZD approach with respect to the generated bits and investigates its performance in the presence of information loss due to FRC events.

4.2.1 Generation of FDb and RPD bits for a z^{th} zone's relays

This sub-section explains the allotment of the *FDb* and *RDb* bits of the relays for both meshed and radial feeders' network. To explain this, a sample network, containing a mesh-loop and two radial feeders, is taken as shown in Figure 4.1. In the figure, due to the bidirectionality in the presence of DGs, a fault current can flow either in Clockwise-direction (C) or Counter-clockwise-direction (CC). A ‘C and CC’ method has been presented to assign the *FDb* to the feeders’ relays, which is explained below.

4.2.1.1 Generation of FDb

The FDb can be (+1) or (-1) and are shown within parenthesis. The generation of FDb is explained by using the Figure 4.1.

1. First, start from a node of any zth zone of the mesh-loop.

2. Then, start assigning (+1) *FDb* to all relays that CC-direction points towards mid of their associated feeder zones.

3. Then, start assigning (-1) *FDb* to all relays that C-direction points towards mid of their associated zones.

4. Choose the next adjacent mesh-loop (if any), and repeat the steps from 1 to 3 for all the unassigned relays in the selected loop.

5. Repeat steps 1 to 4 for all loops.

6. Similarly, repeat steps 2 to 3 for all radial zones, if any.

7. Represent all relays, assigned with (+1) and (-1) [5]*FDb* respectively as *R(zu)* and *R(zd)*i.e. relays at the ‘u’ and ‘d’ ends of a zth zone.

It can be observed from Figure 4.1 that after *FDb*’s allotment, each zone will have a pair of *FDb* of opposite signs and opposite directions facing inwards to the mid of zone.

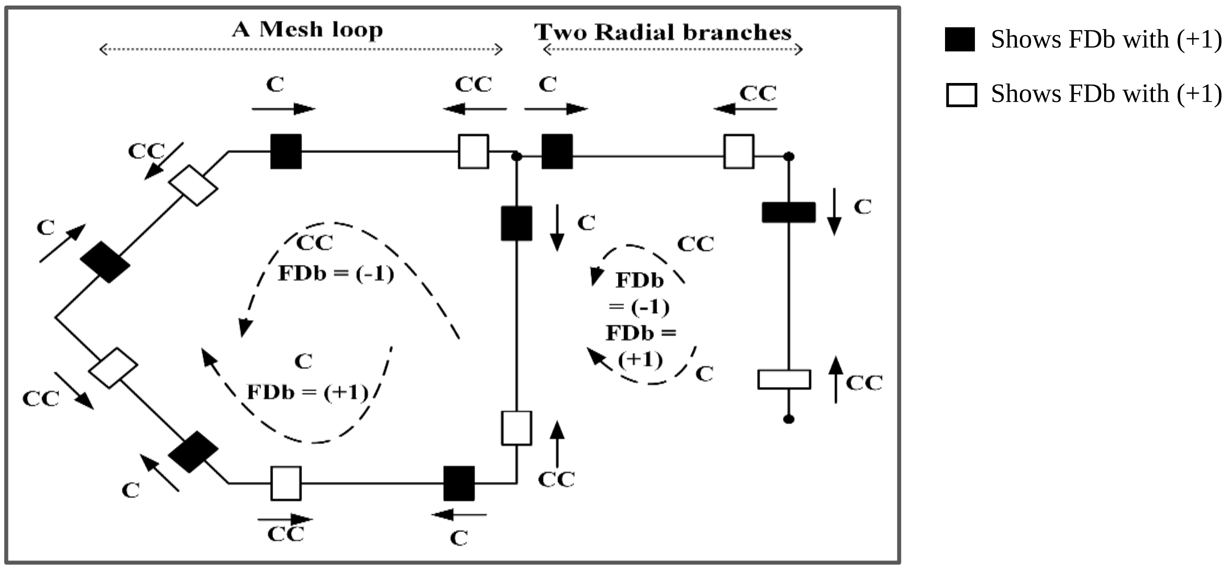


Figure 4.1 Generation of the FDb for a sample radial-meshed network

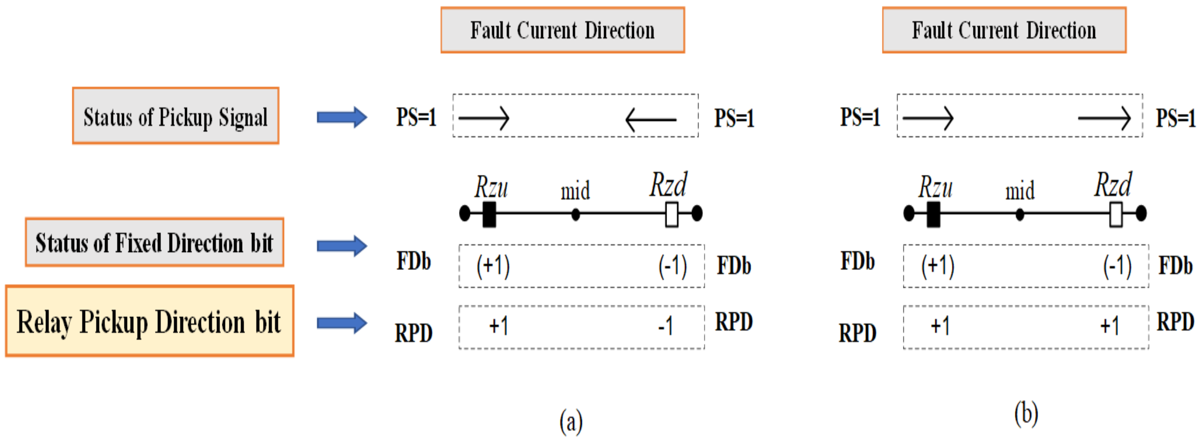


Figure 4.2: Generation of RDb:

a) When, fault current's direction is the same of the direction of FDb
(b) When, fault current's direction is opposite of the direction of FDb

4.2.1.2 Generation of RDb

When there is no fault in the system, the default status of *RDb* will be '0' as *PS* will be zero from Equation (3.3). Now, as discussed, the default direction of a *FDb* bit is towards mid of the zone. So, during a fault, when a relay will experience a fault current with *PS*=1, that direction is the same as the direction of the associated reference *FDb*, then *RDb* bit will be equal to the *FDb* and will be calculated by ***RDb*=(*FDb*×*PS*)**, as shown in Figure 4.2(a). On the other hand, if the fault current's direction is opposite of the direction of *FDb*, then the *RDb* bit will be calculated by ***RDb*=((-*FDb*) ×*PS*)**. This can be seen in Figure 4.2 (b).

4.2.2 Existing FZD methods and Problem formulation

The fundamental logic adopted by the existing FZD methods [20], [60], [61], [93] is similar to the fault detection approach of the differential protection scheme. This fundamental logic is depicted as follows: "when the directions of both end-relays point towards the mid of the zone, only then the CPC will declare the presence of a fault at the zth zone and will turn high the fault-status for the zth zone i.e. *FSz*=1. Otherwise, it will show *FSz*=0 which means that there is no fault at the zth zone."

To formulate the problem, this existing method is depicted by using the generating bits, which is as follows. So, when a relay will sense *PS*=1 status along with directions facing towards the mid of the zone, then as per the generating rule explained in Section 4.2.1, it will turn the magnitude of *RDb* high i.e. *RDb*=1, and then will trigger it communication unit to send the high *RDb* status to the CPC. While, in the normal conditions, it doesn't trigger its communication unit to send the *RDb* signal, as a result, the CPC will hold the default status of *RDb* which is zero i.e. *RDb*=0, and will declare that the FSz=0. Thus, only when the CPC will receive high RDb from both end-relays of a zth zone, then it will declare *FSz*=1. While in the normal balanced situations, the CPC holds the default low *RDb* bits for both end-relays which declares *FSz*=0. Thus, in both these cases, as shown in the first two rows of Table 4.1, the existing FZD methods will identify the fault status correctly. But there are some more cases which can be possible due to the presence of FRC events and abnormal conditions. These cases are shown in the rest of the cases of Table 4.1, which are not considered by the existing FZD methods. Consequently, the existing logic may interpret the situation wrongly. This is explained in detail as follows. In Table 4.1, in the 3rd and 4th cases, one of the end-relays may get failed due to FRC event and, as a result, will show low *RDb* instead of high *RDb*. Resulting of which, the CPC will interpret that the fault has not occurred at the z^{th} zone, while it has happened somewhere else as CPC has got a high *RDb* from the system. Now, in the 5th case when both end-relays fail due to FRC and show low *RDb*s, the CPC may take it as the first case and may interpret that the short circuit fault current is not flowing in this z^{th} zone, which is a wrong identification. Thus, the existing methods fail to detect the faulted zone in the presence of missing information due to FRC events. Furthermore, such existing FZD approaches have been

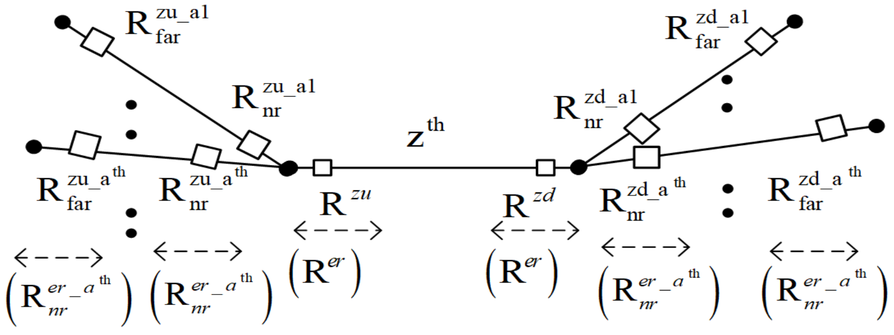


Figure 4.3 A typical feeder network of a z^{th} zone

designed by assuming the cases where the fault currents enter from both of the z^{th}'s ends. Whereas there can be a case where fault current enters only from one z^{th}'s end while doesn't enter from another z^{th}'s end as there is no fault current source beyond this z^{th}'s end. In this situation, as described in the 6th case of Table 4.1, the existing FZD methods fail to recognize the fault in the z^{th} zone. Besides these cases, if due to any non-faulty condition, the *RDb* of the end-relays of a z^{th} zone gets turned high, as shown in the 7th case, then the CPC interprets this situation as a faulty situation. Thus, the existing methods will fail to discriminate the non-fault condition from the fault one.

4.3 Proposed Direction based Fault Detection Method (DFZD)

A direction-based FZD (DFZD) algorithm, namely (DFZD-algo), is proposed for detecting the faulted zone after getting the high *PS* and *RDb* signals from the system relays. In this section, first, the representation of the network with relays and central units, is discussed, then the DFZD-algo is explained.

Table 4.1

FSz detection by the existing FZD method in possible normal and abnormal situations

S.No.	Actual *FSz*	Received *RDb* at the RCU		*FSz* identification by the existing method	Correct/Wrong Identification
		First R^{er}	Second R^{er}		
1.	0	0	0	0	Correct
2.	1	1	1	1	Correct
3.	1	1	0	0	Wrong
4.	1	0	1	0	Wrong
5.	1	0	0	0	Wrong
6.	1	1	*0	0	Wrong
7.	0	1	1	1	Wrong

4.3.1 *Representation of a typical Network*

- A typical network can be illustrated as the interconnections of different R^{er} of different feeder zones, where it can be configured as a mesh, radial or a mixed mesh-radial network configuration. In this work, both ends of a z^{th} feeder zone have been denoted by 'u' and 'd' ends, where Rzu and Rzd are the respective end-relays. Each R^{er} can have a number of adjacent feeder zones containing corresponding adjacent near and far relay as shown in Figure 4.3. Where in the figure, ($R_{nr}^{zu\_a^{th}}$ and $R_{far}^{zu\_a^{th}}$) and ($R_{nr}^{zd\_a^{th}}$ and $R_{far}^{zd\_a^{th}}$) are the near and far relays located at the a^{th} adjacent-zones of the Rzu and Rzd end-relays respectively. If a R^{er} has no adjacent zone, then that end will be called as open-end (OE). If during normal or restoration condition, feeder re-configuration takes place by switching of the connecting-positions of relays, then each associated relay will immediately send their modified position to the CPC so that CPC can be updated with the current network's configuration or with adjacent zones of each end-relay. In this study, a CPC is used for collecting and processing the online data from the system.

- If the network is significantly large, then to avoid the communication latency due to collecting data from the number of distributed relays, the network can be divided into several areas having their own regional protection and control unit (RCU), in order to collect and utilize the information efficiently [56].

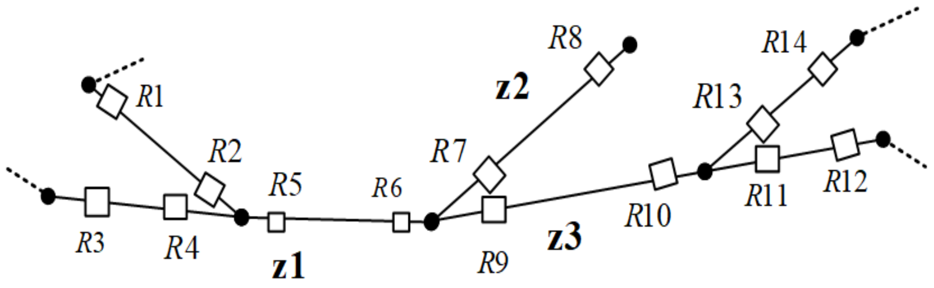


Figure 4.4 A region for protecting three zones upto B1 hierachy level

- An example of a region of three zones, as shown in Figure 4.4, is taken to explain this. In this regional area, besides all three zones (z1, z2, and z3), it includes all the adjacent zones of each zone for the backup protection for which the settings need to be devised. In this example, all the relays of the selected region i.e. (R1 to R14) are assumed to equipped with the communication facilities with its local RCU.

4.3.2 Functioning of the DFZD-algo and Case Studies

The DFZD-algo is based on only three bits (0, +1, -1) and works on the few addition or subtraction of these bits. At RCU, the default status of *RPD* is ‘0’ during normal conditions. Now, as an abnormal condition (faulty or non-faulty) occurs, RCU receives RPD bits (+1 or -1) from the relays. Then, at the RCU, by using *RDb* and allotted *FDb* bits, the DFZD-algo conducts the analysis of the patterns of directions of fault currents in which it assigns the different types of direction status (DS) to different end-relays and their associated different adjacent zones to detect whether the *FSz*=0 or 1. This method eliminates the confusion whether the low *RDb* is either due to normal condition, non-faulty condition, or due to FRC event and thus overcomes the associated drawbacks of the existing FZD methods in detecting the fault location in the presence of FRC events and abnormal conditions. Thus, to identify the correct fault situation in a z^{th} zone or *FSz*, unlike the existing FZD methods, the proposed FZD method doesn’t confine itself to the fault current directions of only the end-relays of the z^{th} zone, but also expand itself to the directions of the fault currents at the adjacent zones.

Some variables and nomenclatures used in this algorithm have been explained first, which are as follows:

DS^{er}: a variable that denotes the DS determined for the selected R^{er} of the z^{th} zone. For example, for Rzu and Rzd, the DS will be DSzu and DSzd respectively.

DS_{ak}: a variable that denotes the DS determined for the kth adjacent zone of the R^{er}.

***x*:** a variable that denotes the *RDb* of a relay, e.g. x^{er} denotes the *RDb* of a R^{er}. While $x_{a^k}^{er,\mathrm{nr}}$ and $x_{a^k}^{er,far}$ denote the RDb of $R_{a^k}^{er,nr}$ and $R_{a^k}^{er,far}$ respectively.

***y*:** a variable that denotes the *FDb* of a relay in the algorithm.

This algorithm generates different types of DS with various nomenclatures ‘IN’, ‘Out’, ‘E’, ‘L’, ‘OE’, ‘All_IN’, ‘Out-1’, and ‘IN_OE’. Where,

“**IN’ and ‘Out’** DS basically denote that the current’s direction inward and outward to an end-node respectively.

‘OE’ i.e. Op_end direction status denotes that there is no flow of the current beyond the associated end, where the reason of the absence of flow can be either absence of the adjacent zones to contribute in the current flow paths or absence of the sources to contribute in supply the current.

‘All_IN’ DS denotes that the current is entering to the zth zone through all of the connected adjacent zones of R^{er}.

‘Out-1’ direction status denotes that atleast through one of the adjacent zones of R^{er}, the current

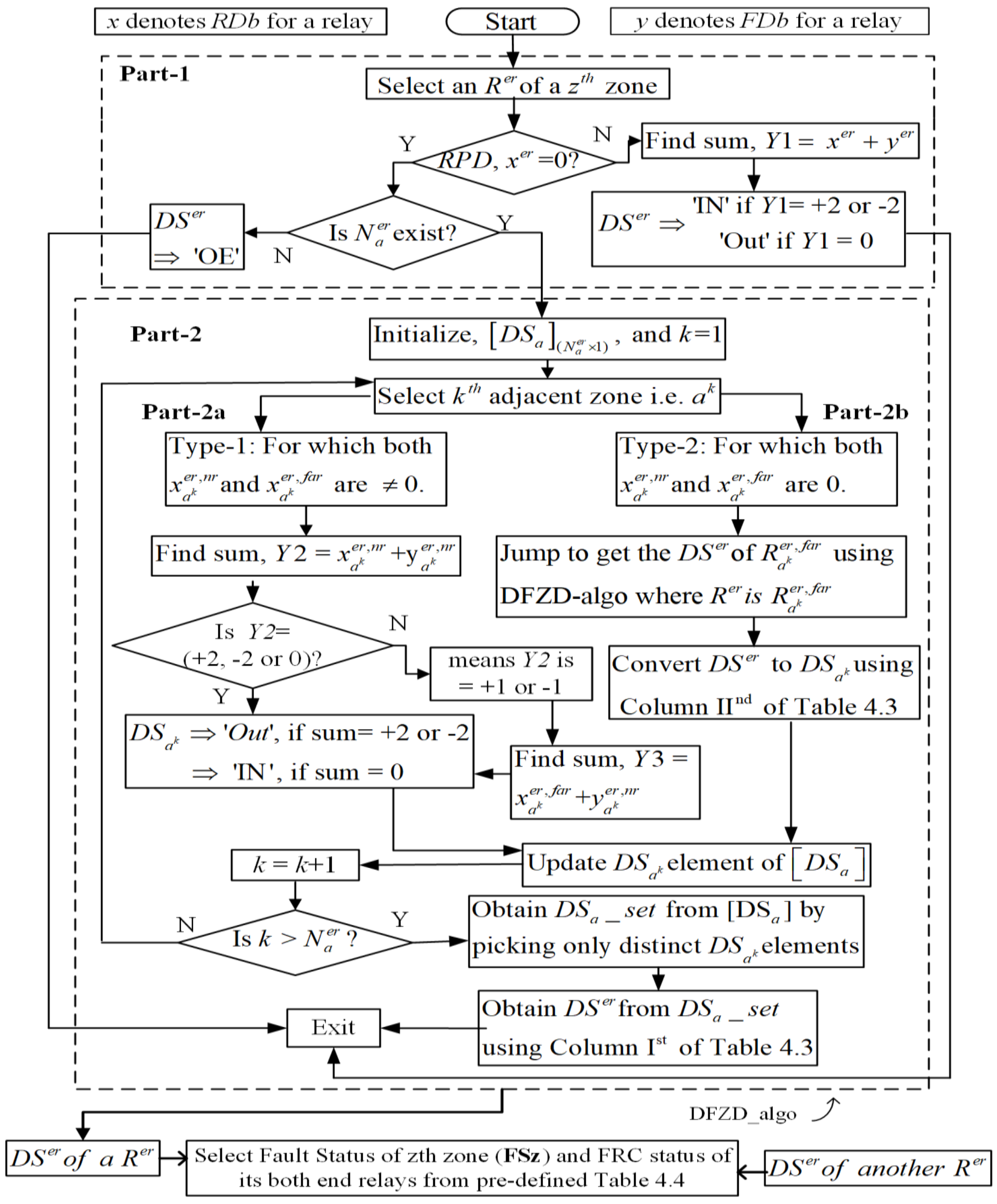
Table 4.2
Possible combinations of the *RDb* of an end-relay with respect to the RDb of near and far relays located at kth adjacent zone

	R^{er}	$R_{a^k}^{er,nr}$	$R_{a^k}^{er,far}$		R^{er}	$R_{a^k}^{er,nr}$	$R_{a^k}^{er,far}$
1.	±1	±1	±1	**5.**	0	±1	±1
2.	±1	±1	0	**6.**	0	±1	0
3.	±1	0	±1	**7.**	0	0	±1
4.	±1	0	0	**8.**	0	0	0

is going outward of the zth zone.

‘IN_OE’ direction status denotes that the current is entering to the zth zone through some of the adjacent zones of R^{er}, while the rest of the adjacent zones are open ended.

The DFZD-algo is designed to find all these possible DS by using *x* and *y* bits associated with the system relays. The algorithm of the DFZD-algo is shown in Figure 4.5. It has two parts, part-1 and part2. The part-1 uses the *x* and *y* of the selected R^{er} to find the DS^{er}. And, if the *x* of R^{er} is detected zero, the algorithm uses its part-2, where it uses the *x* and *y* of the adjacent relays to find the DSak for each of the adjacent zones of the selected R^{er}. Depending upon the normal, abnormal, and FRC situations, the value of *x* can be low or high at the RCU.



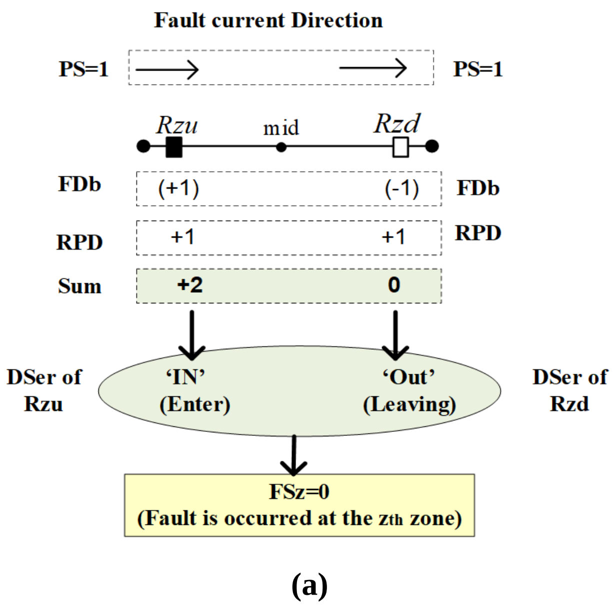
x denotes RDb for a relay
Start
y denotes FDb for a relay
Part-1
Select an R^{er} of a z^{th} zone
RPD, $x^{er}=0$?
Y
N
Find sum, $Y1 = x^{er} + y^{er}$
$DS^{er} \Rightarrow$ 'IN' if $Y1 = +2$ or -2 'Out' if $Y1 = 0$
Is N_a^{er} exist?
$DS^{er} \Rightarrow$ 'OE'
Part-2
Initialize, $[DS_a]_{(N_a^{er}\times 1)}$, and $k=1$
Select k^{th} adjacent zone i.e. a^k
Part-2a
Part-2b
Type-1: For which both $x_{a^k}^{er,nr}$ and $x_{a^k}^{er,far}$ are $\neq 0$.
Type-2: For which both $x_{a^k}^{er,nr}$ and $x_{a^k}^{er,far}$ are 0.
Find sum, $Y2 = x_{a^k}^{er,nr} + y_{a^k}^{er,nr}$
Jump to get the DS^{er} of $R_{a^k}^{er,far}$ using DFZD-algo where R^{er} is $R_{a^k}^{er,far}$
Is $Y2=$ (+2, -2 or 0)?
means $Y2$ is $= +1$ or -1
Convert DS^{er} to DS_{a^k} using Column IInd of Table 4.3
$DS_{a^k} \Rightarrow$ 'Out', if sum= +2 or -2 $\Rightarrow$ 'IN', if sum = 0
Find sum, $Y3 = x_{a^k}^{er,far} + y_{a^k}^{er,nr}$
Update DS_{a^k} element of $[DS_a]$
$k = k+1$
Is $k > N_a^{er}$?
Obtain DS_a_set from $[DS_a]$ by picking only distinct DS_{a^k} elements
Obtain DS^{er} from DS_a_set using Column Ist of Table 4.3
Exit
DFZD_algo
DS^{er} of a R^{er}
Select Fault Status of zth zone (**FSz**) and FRC status of its both end relays from pre-defined Table 4.4
DS^{er} of another R^{er}

Figure 4.5 Proposed algorithm (DFZD_algo) for the detection of zth faulted zone

As a result, the number of cases can appear due to different combinations of x^{nr} and x^{far} with respect to a x^{er}, as shown in Table 4.2, where this table shows that total 8 such cases are possible. In this table, in the first four cases, x^{er} is non-zero, while, in the last four cases, x^{er} is zero. The next sub-section describes how the DFZD-algo detects the faulted zone in all these possible cases shown in Table 4.2.

a) Case-1 to 4: When R^er is working or it has a non-zero RDb

All the first four cases of Table 4.2 represent the case when both end-relays of the z^th zone, for which *FSz* needs to be determined, are in working state. In these situations, the RCU successfully gets the *x* from R^{er} to calculate *Y1*, which is the addition of x^{er} and y^{er} as shown in Figure 4.5. If the mod value of *Y1* is 2, the DS^{er} will be 'IN', otherwise it will be 'Out'. This is explained with the help of examples shown in Figure 4.6. In the example shown in Figure 4.6(a), the algorithm will find 'IN' DS^{er} for both 'u' and 'd' ends of the z^th zone which will declare *FSz*=1 for the z^th zone. Now, in example shown in Figure 4.6(b), the algorithm will



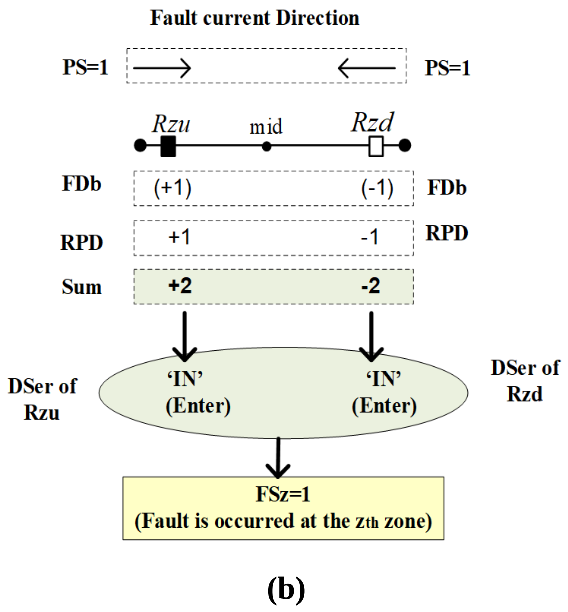


Figure 4.6 Detection of FSz for Case 1 to Case4

find ‘IN’ DS^{er} for ‘u’ end and ‘Out’ DS^{er} for ‘d’ end which, resulting, will declare *FSz*=0 for the zth zone.

b) ***Case 5 and 6: When R^{er} has zero RDb and the reason is unknown***

When a R^{er} has a zero *RDb*, then instead of the absence of flow, it may represent a case where the relay fails to send its *RDb* to the RCU due to FRC. In this situation, as shown in Figure 4.7(a)**,** the conventional differential current based methods such as proposed in [20], [60], [61], [93] may fail to detect the *FSz* or may do the wrong interpretation for *FSz*. While, the DFZD-algo uses the *RDb* of near adjacent relay to determine the DSak of a kth adjacent zone, where it calculates the value of *Y2* which is the addition of $x_{a^k}^{er,\mathrm{nr}}$ and $x_{a^k}^{er,far}$. If the mod value of *Y2* is ‘2’, the DSak will be ‘Out’, as shown in Figure 4.7(b) and if *Y2* is ‘0’, the DSak will be ‘IN’ as shown in Figure 4.7(c).

Now, one more case is possible with ‘IN’ status as shown in Figure 4.7(d). Where if the DFZD-algo finds ‘IN’ DSak status for all connected adjacent zones of the R^{er}, the DS^{er} status of the associated R^{er} relay will be ‘All_IN’ as per algorithm. So, suppose, the Rzd relay is working and its DS^{er} (DSzd) status is ‘E’ or ‘OE’, while for another end-relay Rzu, the algorithm finds All_IN DS^{er}, then the FSz status will be 1, as shown in Figure 4.7(d). Similarly, if the algorithm finds All_IN DS^{er} for both R^{er} s, the *FSz* status will be the same i.e. FSz=1, as shown in Figure 4.7(e). In this way, the DFZD-algo will find the correct status of the *FSz* in the presence of information loss due to FRC event. In the other possible cases, as shown in Figure 4.8(a) and Figure 4.8(b), if atleast one of the adjacent zones has ‘Out’ DSak status, then

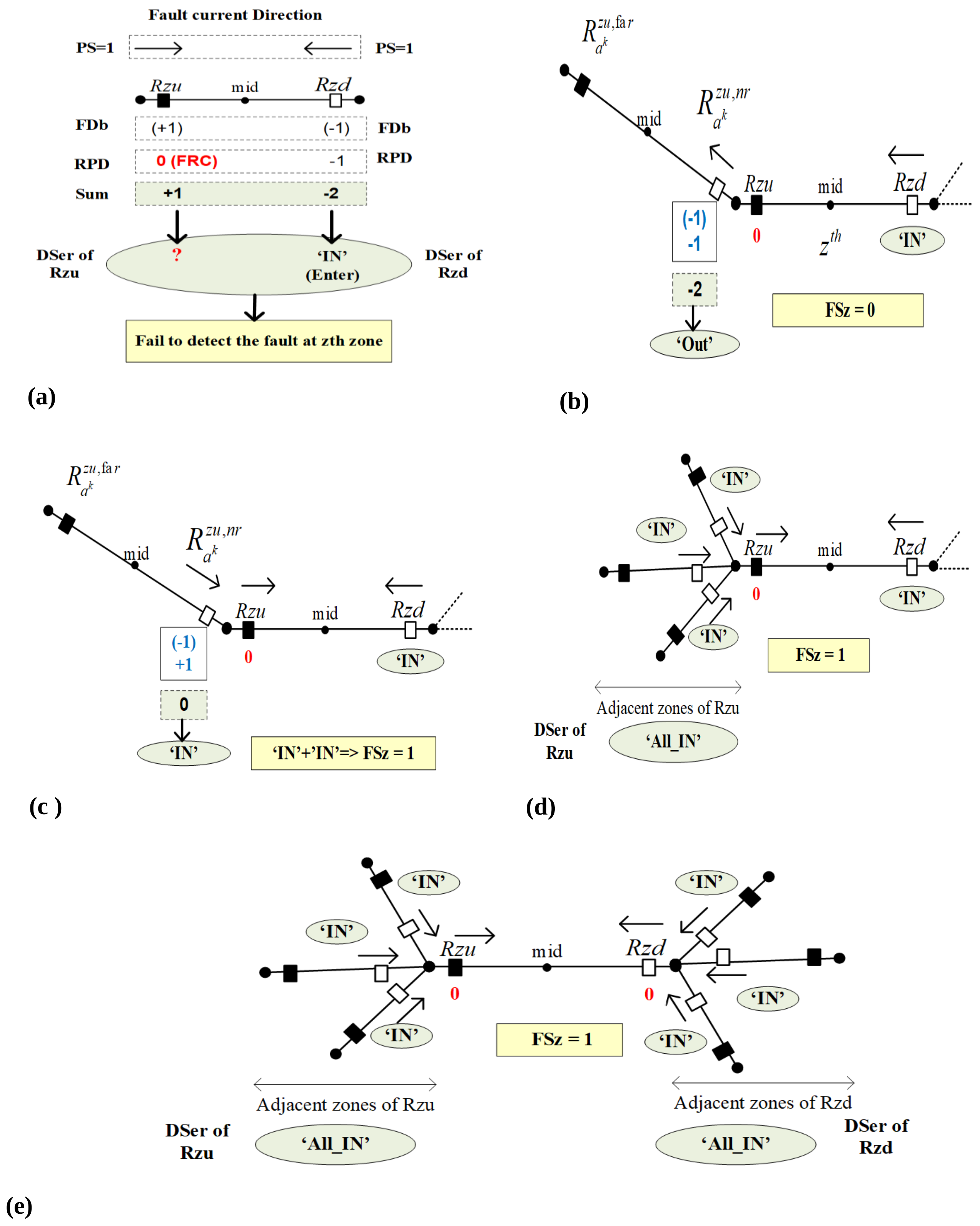


Figure 4.7 Detection of FSz for Case 5 and Case6

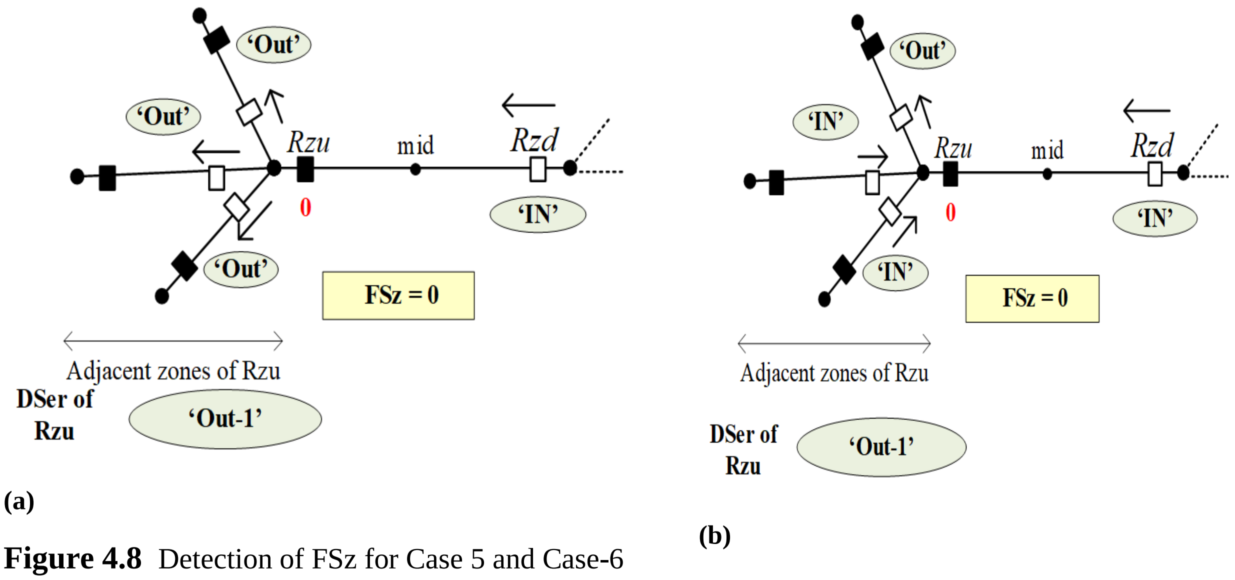


Figure 4.8 Detection of FSz for Case 5 and Case-6

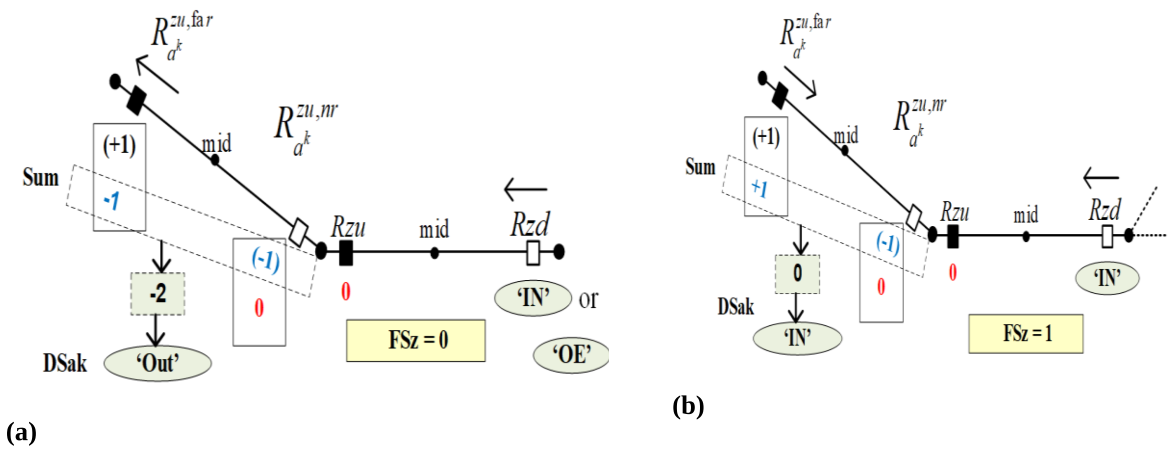


Figure 4.9 Detection of FSz for Case 7

the DS^{er} status of the related R^{er} will be 'Out-1'. The presence of '**Out-1**' direction status for either one of the R^{er} denotes the absence of fault at the zth zone i.e. FSz=0.

c) ***Case 7: When RDbs of both*** R^{er} ***and*** $R_{nr}^{er\_a^{th}}$ ***relays are zero and the reasons are unknown***

This situation may occur due to anyone of the following reasons: 1. When both zth and its adjacent zones are unaffected from the fault, 2. When both relays are failed with FRC. In this situation, when the direction status of both R^{er} and $R_{nr}^{er\_a^{th}}$ relay is not available, the DFZD-algo uses the *RDb* of $R_{far}^{er\_a^{th}}$ relay to determine the actual direction of the fault current's flow. For this, it calculates the value of *Y3* which is the addition of $y_{a^k}^{er,nr}$ and $x_{a^k}^{er,far}$, as shown in flowchart Figure 4.5. Similar to *Y2*, if the mod value of *Y3* is 2, then the DSak status will be

Table 4.3
Conversion Table

Elements in DSa_set	**Column-I**	**Column II: Conversion of Column-Ist's *DS^{er}* to *DS_{ak}***
	DS^{er}	***DS_{ak}***
All elements are 'IN'	All_IN/Enter	IN
Atleast one element is 'Out'	Out-1/L	Out
All elements are 'OE'	OE	OE
Some are 'IN' and some are 'OE'	IN_OE/E	IN

'Out', otherwise if it is 0, then the DSak will be 'IN', as explained in Figure 4.9 (a) and Figure 4.9(b) respectively.

d) ***Case-8: If RDbs of all R^{er}, $R_{nr}^{er\_a^{th}}$, and $R_{far}^{er\_a^{th}}$ are zero and the reasons are unknown***

This is a very rare case; however, it cannot be ignored. To determine the actual reason behind the low RDbs in this case, the DFZD-algo runs its Part-2b where it, first, will find the DSak status for the far-end adjacent relay and then it determines the DS^{er} of the R^{er} with the help of developed Conversion-Table in Table 4.3. The detail description can be found in the next Section (4.3.3).

Discrimination between the non-faulty and faulty conditions:

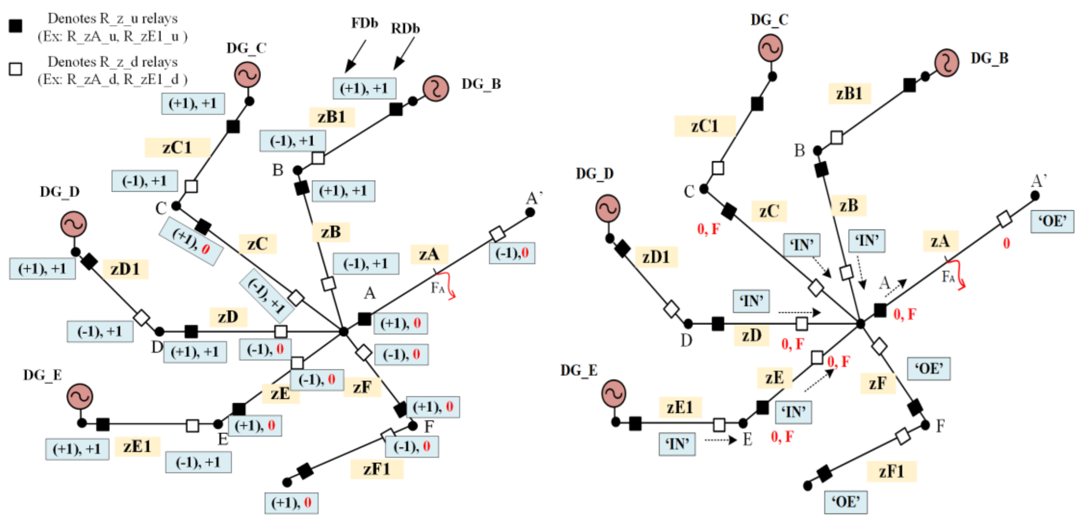
Thus, in this method, an RCU unit parallelly finds *FSz* for each of the associated regional feeder zones upon receiving the high *RDb* bit from the system’s relays. The event of turning the RDb high may happen either due to a non-faulty condition or a faulty condition. If the RCU doesn’t find FSz=1 for any regional zone, then it will reveal that the relays were get picked up due to the non-faulty condition mistakenly. On the other side, if the RCU finds a faulted zone, then it will reveal that the relays were get picked up due to a faulty condition. As the RCU will detect the fault occurrence, it starts determining the relays’ *TMS* settings by using the protection scheme presented in the following Chapter 6, and sends them to all the desired regional primary and backup relays.

4.3.3 Application of the DFZD method

To explain the application of the DFZD-method, an example of a region of six zones, zA, zB, zC, zD, zE, and zF containing four DGs (DG_B, DG_C, DG_D, and DG_E), as shown in Figure 4.10, is taken. This section explains that how the DFZD_algo finds the correct FSz of

Table 4.4: FSz and FRC Determiner Table

DS^{er} of R_z_u with RDb=0	**All_IN**	**OE**	**Out-1**	**IN**	**Out**	**IN-OE**
DS^{er} of R_z_d with RDb=0						
All_IN	1, F(u,d)	1, F(d)	0, F(u,d)	1, F(u,d)	0, F(d)	1, F(d)
OE	0, F(u)	0, *NF	0, NF	1, NF	0, NF	0, F(d)
Out-1	0, F(u,d)	0, NF	0, F(u,d)	0, F(d)	0, F(d)	1, F(d)
IN	1, F(u)	1, NF	0, F(u,d)	1, NF	0, NF	1, F(u,d)
Out	0, F(u)	0, F(u)	0, F(u)	0, NF	**NA	0, F(u,d)
IN-OE	1, F(u)	0, F(u)	1, F(u)	1, F(u,d)	0, F(u,d)	1, F(u,d)


Denotes R_z_u relays (Ex: R_zA_u, R_zE1_u)
Denotes R_z_d relays (Ex: R_zA_d, R_zE1_d)
FDb
RDb
DG_C
DG_B
DG_D
DG_E
zA
zB
zB1
zC
zC1
zD
zD1
zE
zE1
zF
zF1
'IN'
'OE'
0, F

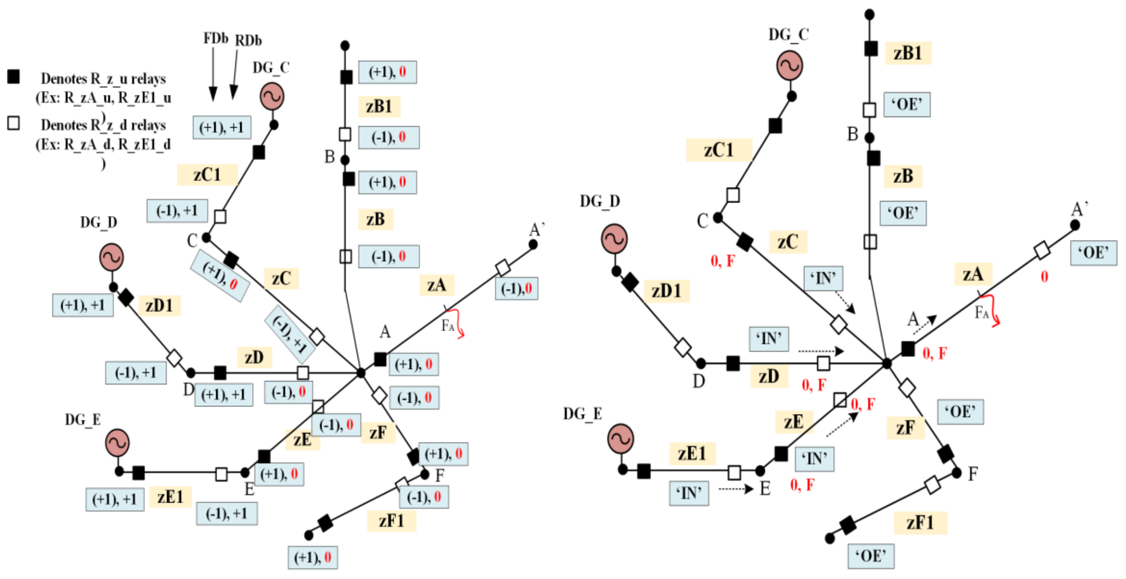
(b) FDb and RDb (all zero RDbs are shown in red font)

(a) Direction Status and failed relays as outputs of DFZD_algo (all RDbs that got zero due to FRC are shown in red font.

Figure 4.10 Detection of FSz in a typical network when all DGs are connected

the zA zone that 'u' end is connected with five adjacent zones (zB, zC, zD, zE, and zF). To show the effectiveness of the DFZD_algo, a typical case has been taken, where all the possible cases as shown in Table 4.2 are included. In the Figure 4.10(a), the RDb bits associated with each adjacent zone make different combinations with respect to the RDb of R_zA_d end-relay as taken from Table 4.2. Where zB, zC, zD, and (zE and zF) represent case 5, 6, 7, and 8 respectively.

In this example, some relays are failed due to FRC event unknowingly. By using this example, it has been explained that how the DFZD method detects the FSz status of the zA zone in the presence of more than one FRC events, and how it differentiates whether the low *RDb* of a relay is due to the absence of fault current, FRC event, or due to open end.



(a) FDb and RDb (all zero RDbs are shown in red font)

(b) Direction Status and failed relays as outputs of DFZD_algo (all RDbs that got zero due to FRC are shown in red font.

Figure 4.11 Detection of FSz in a typical network when DG_B is disconnected

Now, as a fault occurs, the RCU receives *RDb* bits as shown in Figure 4.10(a). Then, by using these available bits, the DFZD_algo finds the DSak status for all adjacent zones (zB, zC, zD, zE, and zF) which are shown in Figure 4.10(b). Where, to determine the DSak status of zB, zC, and zD zones, the DFZD-method uses Part-2a of the DFZD_algo, while for zE1 and zF, it uses Part-2b. Where, before determining the DSak status for the adjacent zones zE and zF that associated both RDb bits are zero, the DFZD_method performs the two sequential operations. In the first operation, it finds the DSak status of their adjacent zones (which are zE1 and zF1 respectively) separately by using the Part-2a of the DFZD_algo. Where the DFZD_algo finds 'IN' and 'OE' DSak outcomes for zE1 and zF1 respectively as shown in Fig(Q4b). As a result, the DS^{er} status of R_zE1_u and R_zF1_u far relays will be 'All_IN' and 'OE' respectively as per the column-I of the Table 4.3. Then, in the second operation, corresponding to these DS^{er} status, the DSak status of the zE and zF will be 'IN' and 'OE' as per the given column-II of the Table 4.3. Similarly, the DFZD_algo, parallely, will determine the DSak of rest of the adjacent zones (zB, zC, and zD) and then will finally yield the outcome in which DS^{er} status of the R_zA_u will be 'IN_OE'. At the same time, the DFZD_method will determine the DS^{er} of the R_zA_d which will be 'OE' in the absence of adjacent zones. Then, by using the developed Determiner-table as shown in Table 4.4, the DFZD_method will yield the *FSz* and FRC status of the zA zone. Where, as per this table, for 'IN_OE' and "OE' DS^{er} status of end_relays, the value of *FSz* will be 1 while F(0,u) is detected which means that u end_relay of the zA zone i.e. R_zA_u is detected failed due to FRC. Now, in another example, as shown in Figure 4.11**,** suppose fault occurs when the DG_B is disconnected, then RCU will receive changed *RDb* bits for zB zone while same *RDb* pattern for the rest of the zones, as shown in Figure 4.11(a). With these RDb inputs, the DFZD_algo will find 'OE' DSak for the zB zone. Whereas, it will find the same direction status for the other adjacent zones (zC, zD, zE, and zF) as shown in the

Table 4.5
Output of the DFZD method and Running time for different cases

	Unknown Status		Output of DFZD method		Running Time (in sec)
Cases with xer=0	FS of z5	FRC	FS of z5	FRC detected in z5 zone	
Case-5	1	R5u and R5d	1	R5u and R5d	0.0079
Case-6	1	R5u, R5d, and R6d	1	R5u and R5d	0.0075
Case-7	1	R5u, R5d, R6d, and R6u	1	R5u and R5d	0.0029
Case-8	1	R5u, R5d, R6d, R6u, and R29d	1	R5u and R5d	0.0032
Case-8a	0	R5u, R5d, R6d, R6u, and R29d	0	R5u and R5d	0.0055

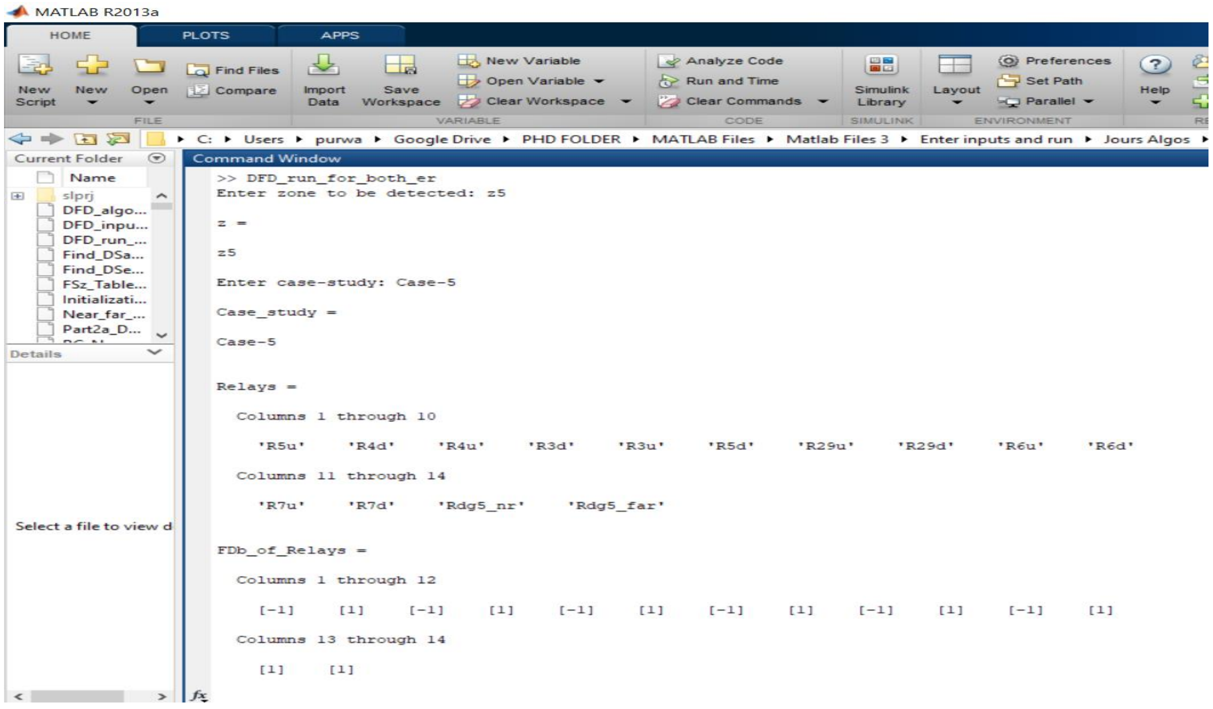
previous example. The final DSak staus are shown in Figure 4.11(b). By using all these DSak, it will find 'IN_OE' DS^{er} for R_zA_u end_relay and 'OE' DS for R_zA_d with *FSz*=1 and F (0, u) final outcomes (by using Table 4.4). Similarly, the RCU will parallelly finds the actual FSz (i.e. FSz=0) for rest of the zones. At the same time, it will also detect the failed relays present at the individual zones. The failed relays are shown in red font in the figures, Figure

4.10(b) and 4.11(b). This explanation shows that the DFZD method works even in the presence of disconnections of the DGs and FRC events.

4.3.4 Results and Discussion

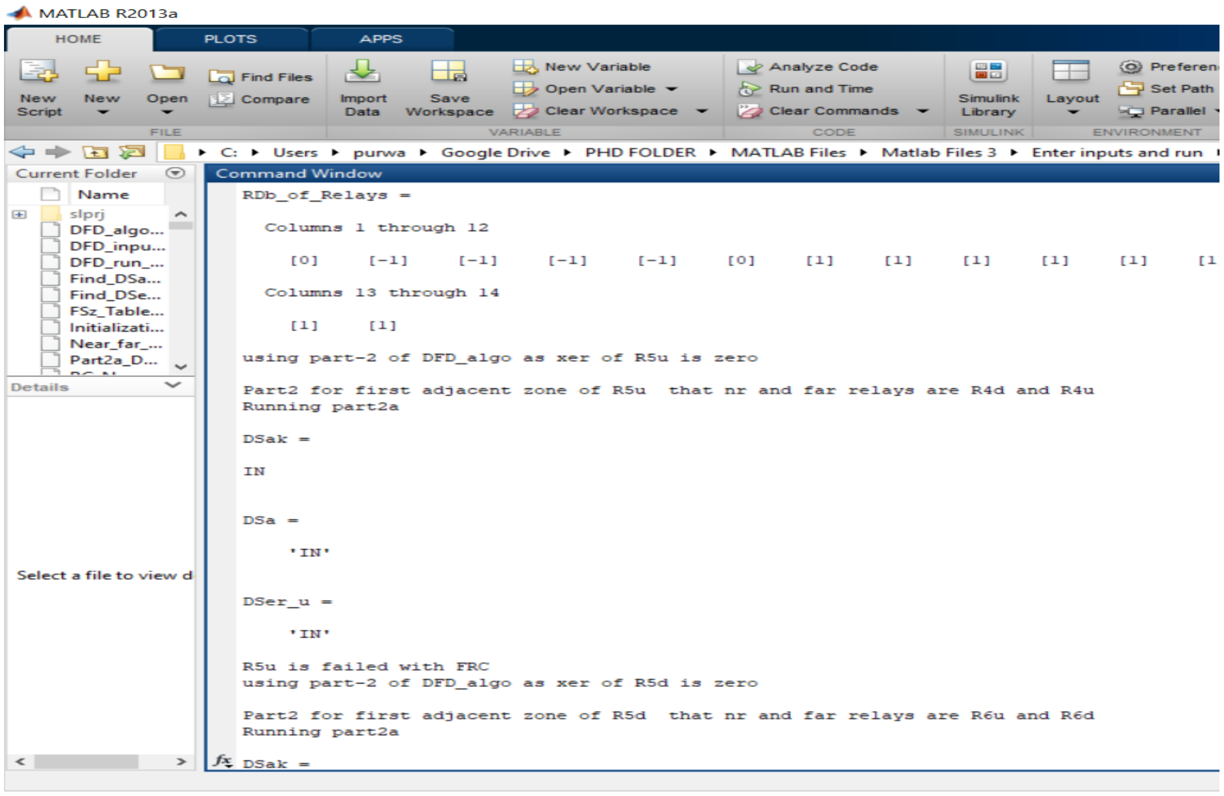
To verify the functioning of the DFZD_algo, matlab coding results for Case-5 to Case-8 (from Table 4.2) for the z5 zone of the Test network (shown in Figure 2.1) have been obtained and discussed. The network relays are assumed to be equipped with the advanced features including the capability of comparing its fault current direction with the assigned *FDb* reference direction (in Section 4.2.1). The relays *PS* signals are calculated by using the Hybrid_algo and the *RDb* bits are determined by using the Section (4.2.1). The presence of the information loss due to FRC events have been taken by making low RDb of the desired relays. All the cases for which the results have been obtained are described in Table 4.5. The results of code running for Case-5 to Case-8a are shown respectively in Figure 4.12 to Figure 4.15. These results verify the performance of DFZD_algo. The key findings of the obtained results are as below:

1. The algorithm successfully finds correct FSz in the presence of more than one unknown FRC event or information loss.
2. It successfully detects whether the end-relays are failed due to FRC event or due to other reasons.
3. It is capable of discriminating the fault condition from the non-fault condition.
4. The proposed algorithm runs very fast in few milliseconds and takes negligible time to detect the fault status and failed relays with FRC. A summary of results can also be seen in Table 4.5.



```
>> DFD_run_for_both_er
Enter zone to be detected: z5

z =

z5

Enter case-study: Case-5

Case_study =

Case-5


Relays =

  Columns 1 through 10

    'R5u'     'R4d'     'R4u'     'R3d'     'R3u'     'R5d'     'R29u'     'R29d'     'R6u'     'R6d'

  Columns 11 through 14

    'R7u'     'R7d'     'Rdg5_nr'     'Rdg5_far'


FDb_of_Relays =

  Columns 1 through 12

    [-1]     [1]     [-1]     [1]     [-1]     [1]     [-1]     [1]     [-1]     [1]     [-1]     [1]

  Columns 13 through 14

    [1]     [1]
```

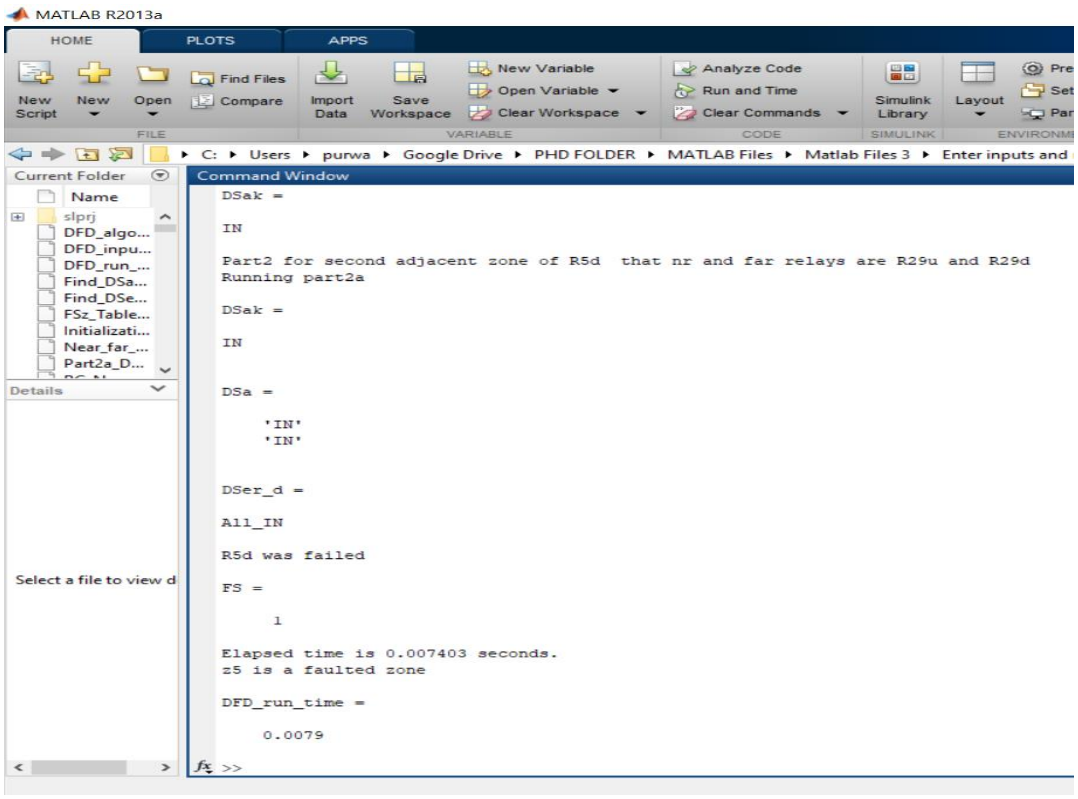
(a)



```
RDb_of_Relays =

  Columns 1 through 12

    [0]     [-1]     [-1]     [-1]     [-1]     [0]     [1]     [1]     [1]     [1]     [1]     [1

  Columns 13 through 14

    [1]     [1]

using part-2 of DFD_algo as xer of R5u is zero

Part2 for first adjacent zone of R5u  that nr and far relays are R4d and R4u
Running part2a

DSak =

IN


DSa =

    'IN'


DSer_u =

    'IN'

R5u is failed with FRC
using part-2 of DFD_algo as xer of R5d is zero

Part2 for first adjacent zone of R5d  that nr and far relays are R6u and R6d
Running part2a

DSak =
```

(b)

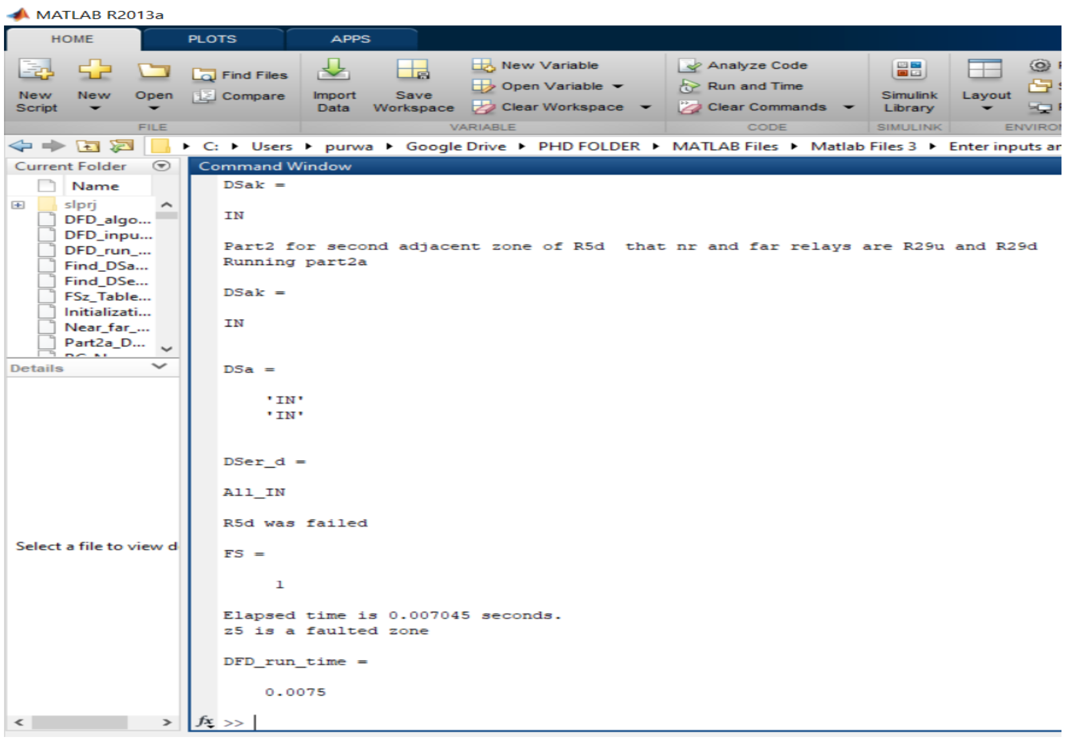
Figure 4.12 Matlab coding outputs for Case-5



```
DSak =

IN

Part2 for second adjacent zone of R5d  that nr and far relays are R29u and R29d
Running part2a

DSak =

IN


DSa =

    'IN'
    'IN'


DSer_d =

All_IN

R5d was failed

FS =

     1

Elapsed time is 0.007403 seconds.
z5 is a faulted zone

DFD_run_time =

    0.0079

fx >>
```

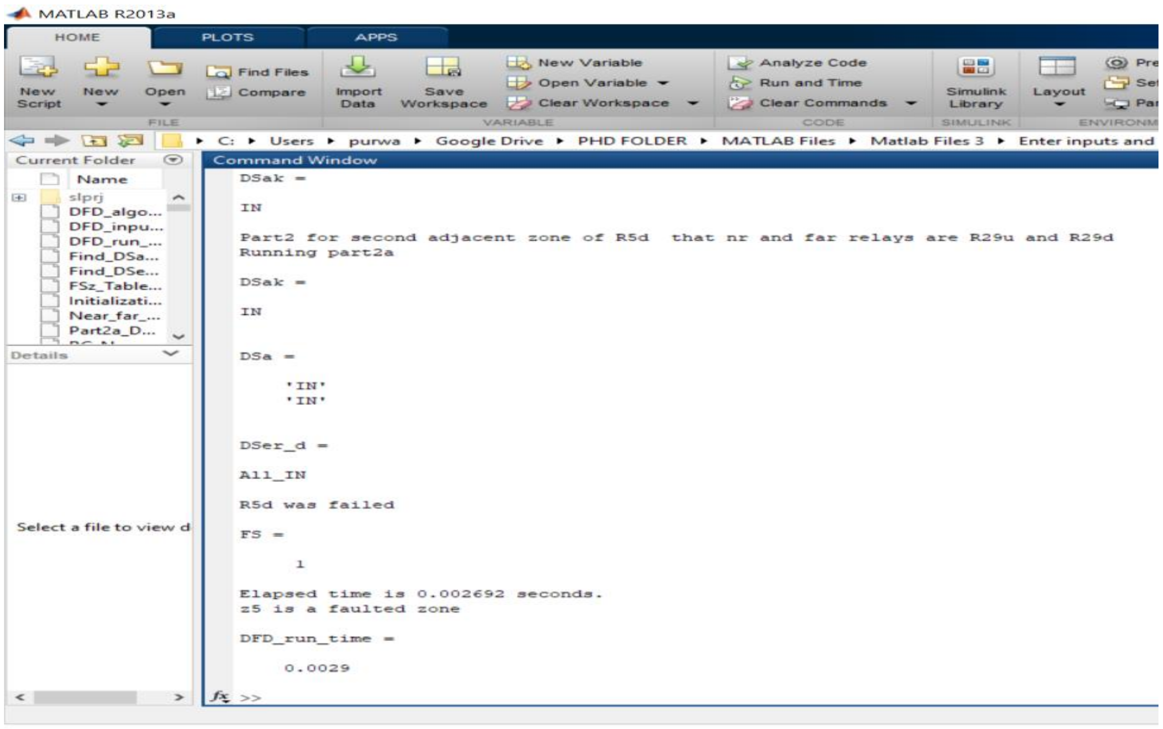
(c)

Figure 4.12 Matlab coding outputs for Case-5



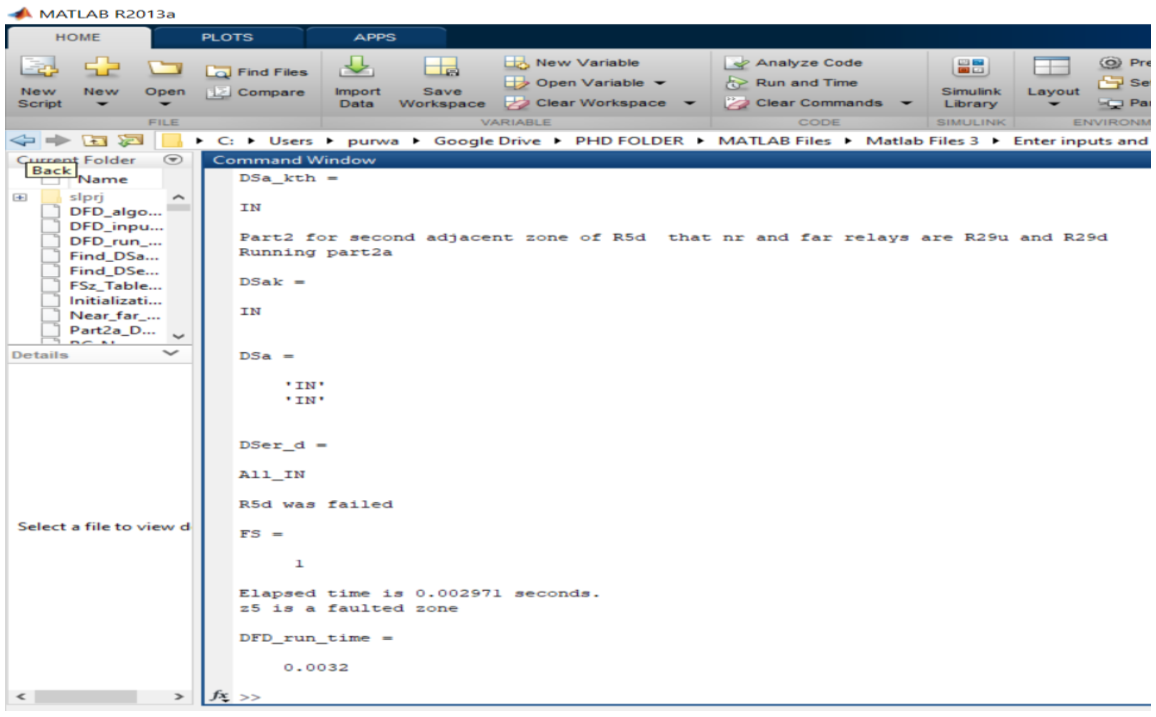
```
DSak =

IN

Part2 for second adjacent zone of R5d  that nr and far relays are R29u and R29d
Running part2a

DSak =

IN


DSa =

    'IN'
    'IN'


DSer_d =

All_IN

R5d was failed

FS =

     1

Elapsed time is 0.007045 seconds.
z5 is a faulted zone

DFD_run_time =

    0.0075

fx >>
```

Figure 4.13 Matlab coding outputs for Case-6



```
DSak =

IN

Part2 for second adjacent zone of R5d  that nr and far relays are R29u and R29d
Running part2a

DSak =

IN


DSa =

    'IN'
    'IN'


DSer_d =

All_IN

R5d was failed

FS =

     1

Elapsed time is 0.002692 seconds.
z5 is a faulted zone

DFD_run_time =

    0.0029

fx >>
```

Figure 4.14 Matlab coding outputs for Case-7



```
DSa_kth =

IN

Part2 for second adjacent zone of R5d  that nr and far relays are R29u and R29d
Running part2a

DSak =

IN


DSa =

    'IN'
    'IN'


DSer_d =

All_IN

R5d was failed

FS =

     1

Elapsed time is 0.002971 seconds.
z5 is a faulted zone

DFD_run_time =

    0.0032

fx >>
```

Figure 4.15 Matlab coding outputs for Case-8

4.4 Conclusion

This chapter proposes an algorithm, named (DFZD_algo), for detecting the faulted zone considering the information loss due to FRC events. This method is independent of the fault current levels which can vary with the change in operating modes. It uses the fault current direction bits collected from the relays for the detection. Where, it takes only a few binary bits from the system's advanced relays and provides the outputs by performing few additions of bits. It works independently of the knowledge of variable operating modes (grid connected mode or islanding mode, network configuration, connections and sizes of DGs) and also independent of the type and level of fault currents. This method helps to make the protection robust and reliable due to its inherent advantageous features, which are as follows: 1. This method is partially independent of the communication systems and works well with the information loss. 2. It is able to differentiate whether the relays get picked up due to a faulty condition or a non-faulty condition. Thus, the proposed method overcomes the associated major drawbacks of existing FZD methods. The functioning of the proposed DFZD_method has been explained by describing different possible cases theoretically and verified by using the coding results. The given explanations and results, concludingly, manifest that this method is an adaptive fault detection approach for a variable DGs-distribution network where the information loss can happen. The results also show that this detection method is fast and takes only few milliseconds to detect the faulted zone and failed relays.