CCITT SGXV
Working Party XV/1
Experts Group for ATM Video Coding

Document AVC-280
June 25 1992

SOURCE : Japan
TITLE : Simplification of FAMC
PURPOSE : Information

This document just informs the content of the contribution from Matsushita which will be submitted to CCITT/MPEG joint session at Angra dos Reis on 6-10 July.

1. Introduction

Field-time Adjusted MC (FAMC MPEG92/24 (AVC-194), MPEG92/100 (AVC-232)) is defined as one of the frame structure prediction mode in TM1draft (Appendix K). The coding performance of FAMC is superior to adaptive field/frame prediction, however the current FAMC specification needs floating calculations. We investigated the simplification of Motion Estimation (ME) and Motion Compensation (MC) of FAMC from the hardware point of view.

This document is concerned to the coding performance of simplified ME and MC for FAMC and accompanyed document (MPEG92/250, AVC-281) describes the hardware complexity of that.

2. Simplified FAMC

Fig. 1 illustrates the principle of FAMC. In current FAMC specification, we need the floating calculation on address generation which is the function of Frame_Distance and *MVfrm*, and interpolation. We simplify FAMC from the hardware point of view. The simplified points of FAMC are:

point 1) N = float(1/(2*Frame_Distance)) is truncated to L1 bit.
point 2) The coefficients for interpolation (a,b) are truncated to L2 bit, and (a+b) is always 2**L2.
point 3) The interpolation results is rounded to the nearest integer at once. (In current specification, interpolation results are truncated three times.)

Point 1 is concerned to address generation and point 2 is concerned to interpolation. Point 3 is just for keeping the mathematical accuracy. In Annex A, the simplified point is shown in the FAMC specification, which is in Appendix K of TM1 draft.

3. Simulation on simplified FAMC

Several simulation was carried out by introducing the above 3 simplification points. These simulation was based on TM0.
(Simulation Conditions)

- Bit rate: 4Mb/s
- N=12, M=3 0-59 frame
- TM0 base, No dual prediction.
- FAMC is introduced for both Luminance and Chrominace prediction.
- MVD: Telescopic search half pixel accuracy (TM1 Appendix K)
- Range of MVD: +-15.5/frame for Flower Garden , +-31.5/frame for Football
- Rate controller: MPEG92/77 step 2 (No modulation)

The results are shown in Figure 2, and will be demonstrated by D1 tape

- In the left part of Fig.2, L1 is changed from 4 to 3 under L2 is 4. In Flower Garden, S/N is dropped when L1 is truncated to 3 bit. Other sequence with L1=3 has the same as L1=4.

- In the center part of Fig.2, L2 is changed from 4 to 1 under L1=4. In all sequence, S/N is dropped if L2 is less than 2.
- Simplified FAMC with L1=4 and L2=3 (FAMC43) has the almost same coding efficiency as original FAMC.(The difference is less than 0.1 dB.)

From those results, we propose to simplify FAMC as

point 1) N = float(1/(2*Frame_Distance)) is truncated to 4 bit.

point 2) The coefficients for interpolation (a,b) are truncated to 3 bit, and (a+b) is always 8.

point 3) The interpolation results is rounded to the nearest integer at once. (In current specification interpolation results truncated three times.)

The simplified FAMC specification is attached as Annex B.

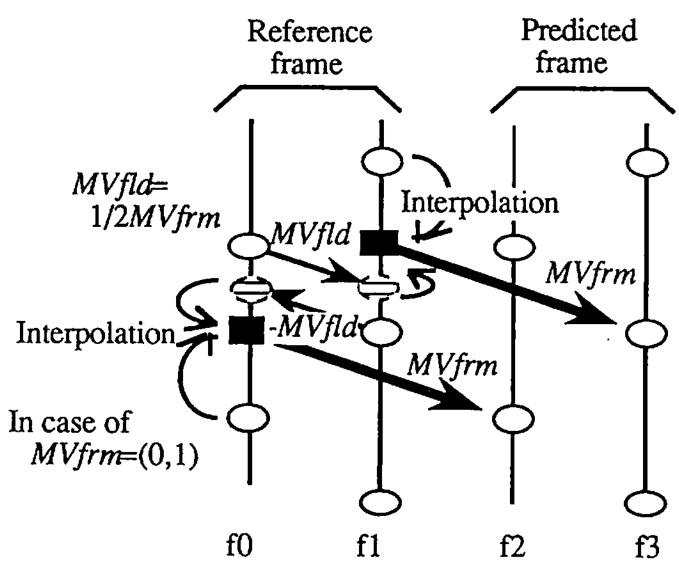

Fig. 1 Principle of FAMC

4. Simplified ME for FAMC

If the vertical component of motion vector is two times of integer (*MVfrm(y)*=2*N), the frame-based prediction and FAMC-based prediction is completely same. It is because, even in the frame-based prediction, the input signal is predicted from the same parity field in this case. It is illustrated in Figure 3. By utilizing this, ME of FAMC is simplified as follows:

Simplified ME of FAMC (Figure 4)

Step 1: frame-based ME with 2(V) x 1(H) accuracy

step 2: simplified FAMC-based ME with +-1.5(V) x +-0.5(H) range and half pixel accuracy around the vector detected in step 1.

5. Simulation on Simplified ME

The simulation with simplified ME and simplified FAMC is carried out at 4Mb/s under the below condition.

(Simulation Conditions)

- Bit rate: 4Mb/s
- N=12, M=3 0-59 frame
- TM0 base, No dual prediction.
- Simplified FAMC is introduced for both Luminance and Chrominace prediction.
- ME: simplified ME (The same L1 or L2 value is used in both ME and MC.)

- MVD: Telescopic search half pixel accuracy (TM1 Appendix K)
- Range of MVD: +-15.5/frame for Flower Garden , +-31.5/frame for Football
- Rate controller: MPEG92/77 step 2 (No modulation)

Simulation result are plotted in the right part of Figure 2 in compared with adaptive field/frame prediction and original FAMC. This result will be demonstrated by D1 tape.

- Simplified FAMC43 with simplified ME43 (L1=4, L2=3) does not drop the coding efficiency in compared with original FAMC.
  (The difference is less than 0.1dB).
- Simplified FAMC43 with simplified ME43 improves the image quality much from adaptive field/frame prediction.

From this results, the performance of simplified ME is same as original ME for FAMC described in TM1.

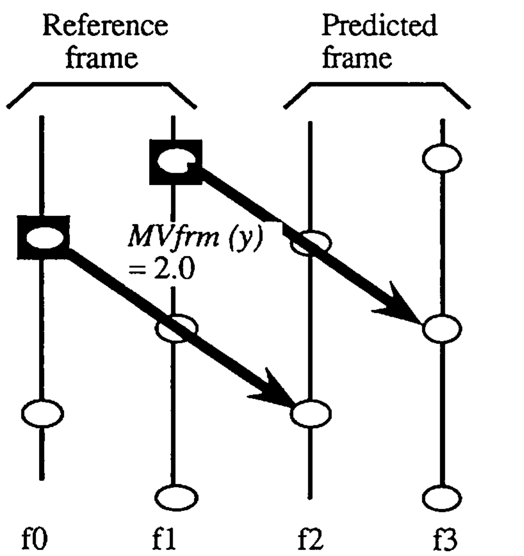

Fig.3 In case of *MVfrm(y)* =2xN frame-base prediction and FAMC prediction are completely same.

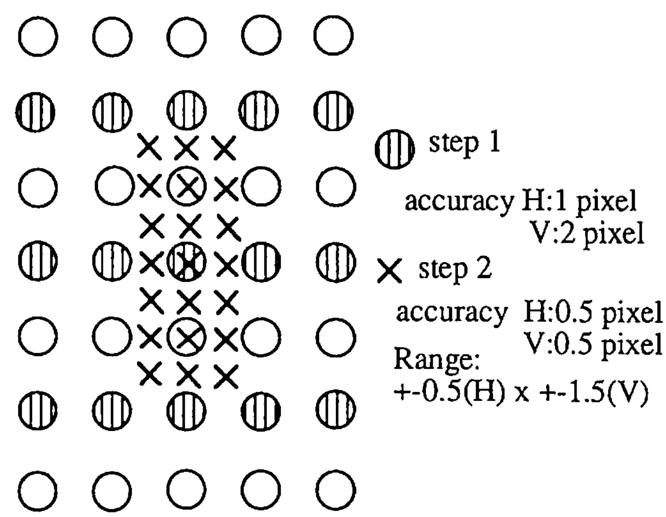

Fig.4 Simplified ME of FAMC

6. Conclusion

The coding performance with simplified FAMC and simplified ME is almost same as the original FAMC. So, ME and MC of FAMC can be simplified without losing its coding efficiency.

Concerning to the hardware complexity, the required operation rate of adaptive field/frame prediction and this simplified prediction were investigated in an accompanyed document (MPEG92/250,AVC-281). The simplified FAMC has about half number of adder operations of adaptive field/frame prediction in ME and MC(Encoder), and the same number in MC(Decoder). And FAMC needs multiplier operations for interpolation.

From those results, we propose to adopt the simplified FAMC43 (L1=4, L2=3) for prediction.

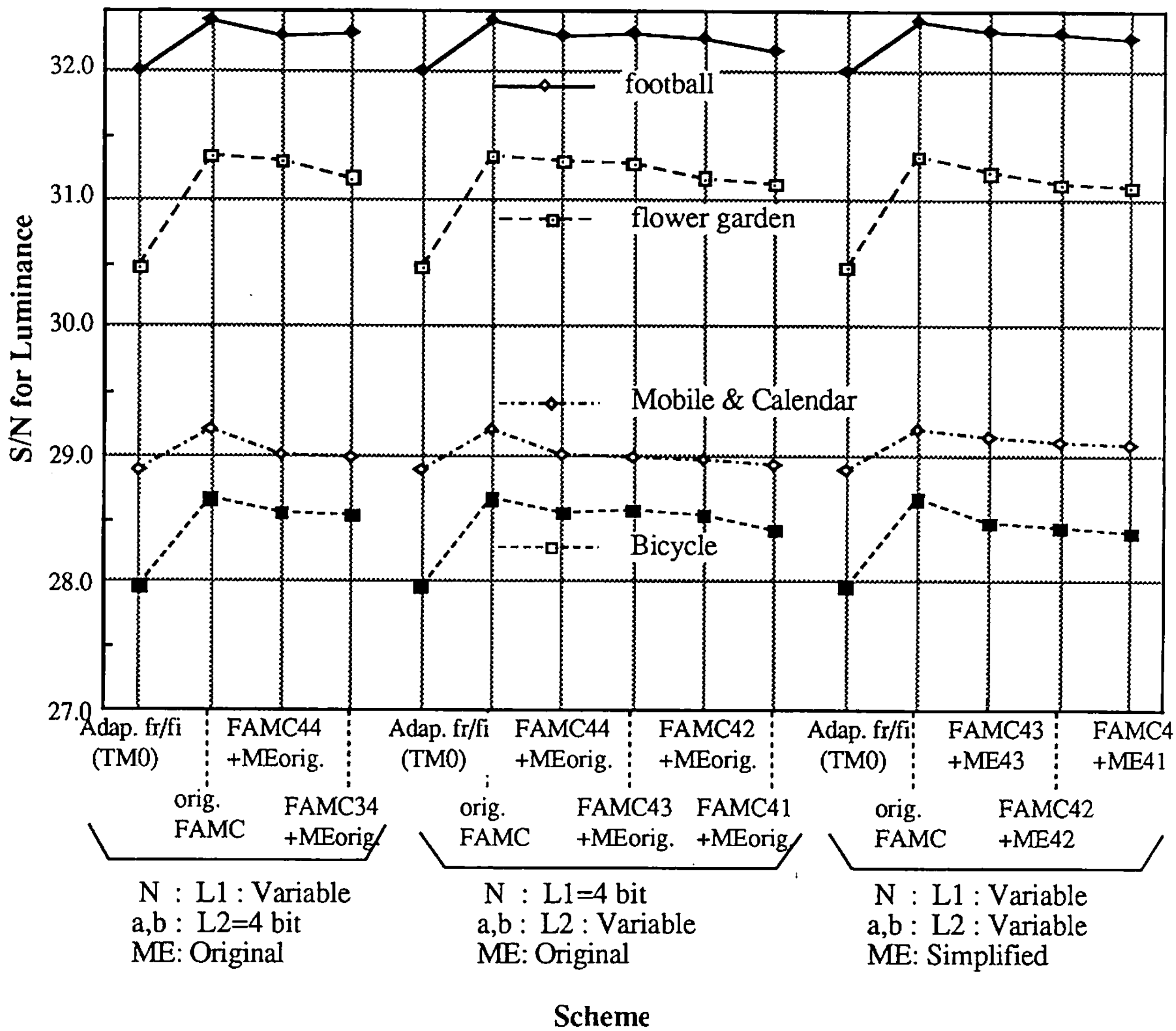

Fig.2 Simulation results of simplification for FA

Annex A Simplified FAMC

```
Get_FAMC_MB_for_Forward (Frame_Distance, Origin, FAMC_MV, FAMC_MB) {

N = float(1/(2*Frame_Distance))    /* =0.5     (0.1000 Bin) Frame_Distance=1 */
                                   /* =0.25    (0.0100 Bin) Frame_Distance=2 */
                                   /* =0.1875  (0.0011 Bin) Frame_Distance=3 */
```

Point 1 N is truncated to L1 bit.

```
for (yl=0; yl<16; ++yl) {
  yg = yl + yorigin
  for (xl=0; xl<16; ++xl) {
    xg = xl + xorigin

    if (yl == even) {              /* For first(Even) Field   */

      x_even1 = xg + (2 * FAMC_MVx)/2
      x_even2 = xg + (2 * FAMC_MVx)//2
      y_even = yg + Adjacent_Even_Line_for_Even_Field_for_Forward
                    (Frame_Distance, FAMC_MVy)

      x_odd1 = xg + ((4 * (FAMC_MVx - N*FAMC_MVx))//2)/2
      x_odd2 = xg + ((4 * (FAMC_MVx - N*FAMC_MVx))//2)//2
      y_odd = yg + Adjacent_Odd_Line_for_Even_Field_for_Forward
                    (Frame_Distance, FAMC_MVy)

                    /* Horizontal and Vertical interpolation   */
      ref_even = ref_frame(x_even1,y_even) + ref_frame(x_even2,y_even)
      ref_odd  = ref_frame(x_odd1,y_odd)   + ref_frame(x_odd2,y_odd)
      FAMC_MB(xl,yl) = (a*ref_even + b*ref_odd)//(2*(a+b))
```

Point 2: a,b is truncated to L2 bit and (a+b) is always 2L2**
Point 3: The interpolation results are rounded to the nearest integer at once.

```
    }

    else {

      < Same as first field >

    }
  }
 }
}
```

ANNEX B The specification of simplified FAMC
==

1. Motion Compensation of simplified FAMC

```
Get_Simplified_FAMC_MB_for_Forward (Frame_Distance, Origin, FAMC_MV, FAMC_MB) {

LeftPel = 0,  RightPel = 703,  TopEvenLine = 0,  BottomEvenLine = 478, TopOddLine = 1,  BottomOddLine = 479
M = 16 // (2*Frame_Distance))       /* M = N * 16; N = 1/(2*Frame_Distance) */
                                    /*   = 8    When Frame_Distance=1  */
                                    /*   = 4    When Frame_Distance=2  */
                                    /*   = 3    When Frame_Distance=3  */

/* For first(Even) Field */
xorigin_even1 = xorigin +(2 * FAMC_MVx)/2
xorigin_even2 = xorigin +(2 * FAMC_MVx)//2
xorigin_odd1  = xorigin + (2*FAMC_MVx - (M*FAMC_MVx)//8)/2
xorigin_odd2  = xorigin + (2*FAMC_MVx - (M*FAMC_MVx)//8)//2
yorigin_even  = yorigin + Adjacent_Even_Line_for_Even_Field_for_Forward
                          (Frame_Distance, FAMC_MVy)
yorigin_odd   = yorigin + Adjacent_Odd_Line_for_Even_Field_for_Forward
                          (Frame_Distance, FAMC_MVy)

for (x1=0; x1<16; ++x1) {
   x_even1 = x1 + xorigin_even1          /* Addressing X of pixel P       */
   x_even2 = x1 + xorigin_even2          /* Addressing X of pixel Q       */
   x_odd1  = x1 + xorigin_odd1           /* Addressing X of R pixel       */
   x_odd2  = x1 + xorigin_odd2           /* Addressing X of S pixel       */

   /* In case that the required pixel is out of frame   */
   if (x_even1 < LeftPel) x_even1 = LeftPel
   if (x_even1 > RightPel) x_even1 = RightPel
   if (x_odd1 < LeftPel) x_odd1 = LeftPel
   if (x_odd1 > RightPel) x_odd1 = RightPel

   for (y1=0; y1<16; y1=y1+2) {
      y_even = y1 + yorigin_even         /* Addressing Y of P & Q pixels */
      y_odd  = y1 + yorigin_odd          /* Addressing Y of R & S pixels */

      /* In case that the required pixel is out of frame   */
      if (y_even < TopEvenLine) y_even = TopEvenLine
      if (y_even > BottomEvenLine) y_even = BottomEvenLine
      if (y_odd < TopOddLine) y_odd = TopOddLine
      if (y_odd > BottomOddLine) y_odd = BottomOddLine

         /* interpolation      */
         ref_even =  ref_frame(x_even1, y_even) + ref_frame(x_even2, y_even)
         ref_odd  =  ref_frame(x_odd1, y_odd)   + ref_frame(x_odd2, y_odd)
         FAMC_MB(x1, y1) = (a*ref_even + b*ref_odd)//16
      }
   }
```

```
/* For Second(Odd) Field */
xorigin_odd1  = xorigin +(2 * FAMC_MVx)/2
xorigin_odd2  = xorigin +(2 * FAMC_MVx)//2
xorigin_even1 = xorigin +(2 * FAMC_MVx + (M*FAMC_MVx)//8)/2
xorigin_even2 = xorigin +(2 * FAMC_MVx + (M*FAMC_MVx)//8)//2
yorigin_odd   = yorigin + Adjacent_Odd_Line_for_Odd_Field_for_Forward
                          (Frame_Distance, FAMC_MVy)
yorigin_even  = yorigin + Adjacent_Even_Line_for_Odd_Field_for_Forward
                          (Frame_Distance, FAMC_MVy)

for (x1=0; x1<16; ++x1) {
   x_odd1  = x1 + xorigin_odd1         /* Addressing X of R pixel      */
   x_odd2  = x1 + xorigin_odd2         /* Addressing X of S pixel      */
   x_even1 = x1 + xorigin_even1        /* Addressing X of pixel P      */
   x_even2 = x1 + xorigin_even2        /* Addressing X of pixel Q      */

   /* In case that the required pixel is out of frame   */
   if (x_odd1 < LeftPel) x_odd1 = LeftPel
   if (x_odd1 > RightPel) x_odd1 = RightPel
   if (x_even1 < LeftPel) x_even1 = LeftPel
   if (x_even1 > RightPel) x_even1 = RightPel

   for (y1=1; y1<16; y1=y1+2) {
      y_odd  = y1 + yorigin_odd        /* Addressing Y of R & S pixels */
      y_even = y1 + yorigin_even       /* Addressing Y of P & Q pixels */

      /* In case that the required pixel is out of frame   */
      if (y_even < TopEvenLine) y_even = TopEvenLine
      if (y_even > BottomEvenLine) y_even = BottomEvenLine
      if (y_odd < TopOddLine) y_odd = TopOddLine
      if (y_odd > BottomOddLine) y_odd = BottomOddLine

         /* interpolation      */
         ref_odd  = ref_frame(x_odd1,y_odd)   + ref_frame(x_odd2,y_odd)
         ref_even = ref_frame(x_even1,y_even) + ref_frame(x_even2,y_even)
         FAMC_MB(x1,y1) = (a*ref_odd + b*ref_even)//16
      }
   }
}
```

For backward motion compensation, the same change as forward prediction should be done from specification in TM1.

(Adjacent_XXX_Line_for_XXX_Field_for_XXX) functions table is same as TM1.
The vertical interpolation coefficients (a,b) are shown in Table 2.1 to 2.2.

Prediction for Chrominance is the same way defined in TM1.

2. Motion Vector Estimation for FAMC

For ME simplification, 2 Step MV search algorithm is used.

1) Step 1 : Integer pel, 2 line accuracy
In step 1, frame-base ME is performed with 2(V) x 1(H) accuracy.

```
Min_AE = MAXINT
for (j=(-YRange); j<(YRange+1); j+=2) {
   for (i=(-XRange); i<(XRange+1); ++i) {
      Get_Prediction_MB_by_Frame_Prediction (i, j, prediction_mb)
      AE_mb = AE_macroblock (current_mb, prediction_mb)
      if (AE_mb < Min_AE) {
         Min_AE = AE_mb
         FAMC_MV = (i,j)
      }
   }
}
```

2) Step 2 : Half pel accuracy

In step 2, simplified FAMC based-ME is performed on the twenty neighboring positions which are evaluated the following order;

```
1   2   3
4   5   6
7   8   9
10  0   11
12  13  14
15  16  17
18  19  20
```

where 0 represents the evaluated position in step 1.
Min_AE as a result of in Step 1 is used as an initial value in Step 2.

```
for (j=-3; j<4; ++j) {
   for (i=-1; i<2; ++i) {
      Get_Simplified_FAMC_MB_for_xxxx (Frame_Distance, Origin,
                  (FAMC_MVx_2int+0.5*i, FAMC_MVy_2int+0.5*j), FAMC_MB)
      AE_famc = AE_macroblock (current_mb, FAMC_MB)
      if (AE_famc < Min_AE) {
         Min_AE = AE_famc
         FAMC_MV = (FAMC_MVx_2int+0.5*i, FAMC_MVy_2int+0.5*j)
      }
   }
}
```

where (FAMC_MV_2int) represents the motion vector which is detected in step 1 motion estimation stage.

Table 2.1 (a, b)
(yl == even) in forward prediction
(yl == odd) in backward prediction

FAMC_MVy	Frame_Distance 1 (a, b)	Frame_Distance 2 (a, b)	Frame_Distance 3 (a, b)
-1.0	3, 5	2, 6	1, 7
-0.5	5, 3	4, 4	4, 4
0.0	8, 0	8, 0	8, 0
0.5	5, 3	4, 4	4, 4
1.0	3, 5	2, 6	1, 7
1.5	1, 7	2, 6	3, 5
2.0	8, 0	8, 0	8, 0
2.5	1, 7	6, 2	5, 3
3.0	3, 5	3, 5	3, 5
3.5	5, 3	2, 6	0, 8
4.0	8, 0	8, 0	8, 0
4.5	5, 3	2, 6	6, 2
5.0	3, 5	3, 5	4, 4
5.5	1, 7	6, 2	2, 6
6.0	8, 0	8, 0	8, 0
6.5	1, 7	2, 6	2, 6
7.0	3, 5	2, 6	4, 4
7.5	5, 3	4, 4	6, 2
8.0	8, 0	8, 0	8, 0
8.5	5, 3	4, 4	0, 8
9.0	3, 5	2, 6	3, 5
9.5	1, 7	2, 6	5, 3
10.0	8, 0	8, 0	8, 0
10.5	1, 7	6, 2	3, 5
11.0	3, 5	3, 5	1, 7
11.5	5, 3	2, 6	4, 4
12.0	8, 0	8, 0	
12.5	5, 3	2, 6	
13.0	3, 5	3, 5	
13.5	1, 7	6, 2	
14.0	8, 0	8, 0	
14.5	1, 7	2, 6	
15.0	3, 5	2, 6	
15.5	5, 3	4, 4	
16.0	Repeated		
16.5			
17.0			

Table 2.2 (a, b)
(yl == odd) in forward prediction
(yl == even) in backward prediction

FAMC_MVy	Frame_Distance 1 (a, b)	Frame_Distance 2 (a, b)	Frame_Distance 3 (a, b)
-1.0	3, 5	2, 6	1, 7
-0.5	3, 5	3, 5	4, 4
0.0	8, 0	8, 0	8, 0
0.5	3, 5	3, 5	4, 4
1.0	3, 5	2, 6	1, 7
1.5	6, 2	5, 3	5, 3
2.0	8, 0	8, 0	8, 0
2.5	6, 2	1, 7	1, 7
3.0	3, 5	3, 5	3, 5
3.5	3, 5	6, 2	5, 3
4.0	8, 0	8, 0	8, 0
4.5	3, 5	6, 2	1, 7
5.0	3, 5	3, 5	4, 4
5.5	6, 2	1, 7	6, 2
6.0	8, 0	8, 0	8, 0
6.5	6, 2	5, 3	6, 2
7.0	3, 5	2, 6	4, 4
7.5	3, 5	3, 5	1, 7
8.0	8, 0	8, 0	8, 0
8.5	3, 5	3, 5	5, 3
9.0	3, 5	2, 6	3, 5
9.5	6, 2	5, 3	1, 7
10.0	8, 0	8, 0	8, 0
10.5	6, 2	1, 7	5, 3
11.0	3, 5	3, 5	1, 7
11.5	3, 5	6, 2	4, 4
12.0	8, 0	8, 0	
12.5	3, 5	6, 2	
13.0	3, 5	3, 5	
13.5	6, 2	1, 7	
14.0	8, 0	8, 0	
14.5	6, 2	5, 3	
15.0	3, 5	2, 6	
15.5	3, 5	3, 5	
16.0	Repeated		
16.5			
17.0			

*In case of FAMC_MVy < 0, all coefficients (a, b) are cyclically repeated in Table 2.1 and 2.2.