# Annex C. Source codes – MATLAB and z-tree

Table 1C lists all MATLAB and "z-tree" source code files that have been used for the simulations, experiments and data evaluations in chapter 3. Afterwards, the source codes are listed. The MATLAB source codes are comprehensive. The "z-tree" codes are only listed as far as they are directly accessible over the graphical interface. All other "z-tree" source code has been omitted.

**Table 1C.** MATLAB and "z-tree" files that have been programmed for the simulations, experiments and data evaluations

| Program | Number | | File |
|---|---|---|---|
| MATLAB | 1. | 1 | cournot_werte_berechnung.m |
| | | 2 | cournot_werte_schnell.m |
| | | | |
| | | 3 | experiment_mass_market_result_round_profits.m |
| | | 4 | experiment_mass_market_result_bar_chart_bilateral.m |
| | | 5 | experiment_mass_market_result_cake_chart_unilateral.m |
| | | | |
| | | 6 | experiment_individual_parameters_rnpvs_expected_events.m |
| | | 7 | experiment_individual_parameters_rnpvs_unexpected_events.m |
| | | 8 | experiment_individual_result_calculations_tables.m |
| | | 9 | experiment_individual_result_low_costs_figure.m |
| | | | |
| z-tree | 2. | 1 | 4-2-2-1-cournot-equilibrium.ztt |
| | | | 4-2-2-2-cournot-negotiations.ztt |
| | | 2 | 5-3-2-1 game dynamic low expected complete.ztt |
| | | | 5-3-2-2 game dynamic low expected incomplete.ztt |
| | | | 5-3-3-1 game dynamic low unexpected complete.ztt |
| | | | 5-3-3-2 game dynamic low unexpected incomplete.ztt |
| | | 3 | 5-3-2-3 game dynamic high expected complete.ztt |
| | | | 5-3-2-4 game dynamic high expected incomplete.ztt |
| | | | |

### 1.1. File: cournot_werte_berechnung.m

```
clear;  syms q1 q2 p real;
a = 20; c1 = 1; c2=1; d=0.9;

Pi1 = '(a-q1-d*q2-c1)*q1';        Pi2 = '(a-q2-d*q1-c2)*q2';
Pi1d= diff(Pi1,q1);               Pi2d= diff(Pi2,q2);
```

```
S = solve(Pi1d,Pi2d);
q1_opt = eval(S.q1);            q2_opt = eval(S.q2);
p1 = a - q1_opt - d*q2_opt;     p2 = a - q2_opt - d*q1_opt;
q_total = q1_opt + q2_opt;
%q1 = (a-2*c1+c2) / 3; %q2 = (a-2*c2+c1) / 3
Pi1_opt = (p1 - c1) * q1_opt;   Pi2_opt = (p2 - c2) * q2_opt;
Pi_total=Pi1_opt+Pi2_opt;
if ((p1<c1) | (p2<c2) | (q1_opt<0) | (q2_opt<0))
   Pi1_opt=0; Pi2_opt=0;
end
%-----------------------------------------------------------------

Pi_cart = '(a-(q1+d*q2))*(q1)-c1*(q1)+(a-(q2+d*q1))*(q2)-c2*(q2)';
Pi1d_cart= diff(Pi_cart,q1);    Pi2d_cart= diff(Pi_cart,q2);
S_cart = solve(Pi1d_cart,Pi2d_cart);
q1_cart_opt = eval(S_cart.q1);  q2_cart_opt = eval(S_cart.q2);
q_cart_total = q1_cart_opt + q2_cart_opt;
p1_cart = a - q1_cart_opt - d*q2_cart_opt;
p2_cart = a - q2_cart_opt - d*q1_cart_opt;

%p_cart = a - q1_cart_opt - q2_cart_opt; %???
Pi1_cart_opt = (a- q1_cart_opt - d*q2_cart_opt - c1) *
q1_cart_opt;
Pi2_cart_opt = (a- q2_cart_opt - d*q1_cart_opt - c2) *
q2_cart_opt;
Pi_cart_total=Pi1_cart_opt+Pi2_cart_opt;
added_value = Pi_cart_total-Pi_total;
added_value_rel = (Pi_cart_total-Pi_total)/Pi_total;

if ((p1<c1) | (p2<c2) | (q1_cart_opt<0) | (q2_cart_opt<0))
     Pi1_opt=0; Pi2_opt=0;
end
```

### 1.2. File: cournot_werte_schnell.m

```
clear;
a = 20; c1 = 1; c2=1; d1=1; d2=0.9;
q1=6.5517; q2=5;
Pi1 = (a- d1*q1 - d2*q2 - c1) * q1;
Pi2 = (a- d1*q2 - d2*q1 - c2) * q2;
```

### 1.3. File: experiment_mass_market_result_round_profits.m

```
clear; scrsz = get(0,'ScreenSize'); figure('Position',[50 100 600
200]); hold on;
verzeichnis='c:\game theory\ergebnisse\phase 2010-2
dissertation\MS Excell experimental data\';
datei ='collected data - mass market.xls';
pfad = [verzeichnis, datei];  N=48;
```

```
dat1 = xlsread(pfad, 1, 'E54');          dat2 = xlsread(pfad, 2,
'E54');
dat3 = xlsread(pfad, 3, 'B3:F241');      dat4 = xlsread(pfad, 4,
'B3:F241');

for lv1=1:N
    lv2 = (lv1-1)*5;
    entscheidung3(lv1,1,:) = dat3(lv2+3,:); entscheidung3(lv1,2,:)
= dat3(lv2+4,:);
    entscheidung4(lv1,1,:) = dat4(lv2+3,:); entscheidung4(lv1,2,:)
= dat4(lv2+4,:);
end
entscheidung3_av_temp=(sum(entscheidung3(:,1,:)+entscheidung3(:,2,
:)))/(N*2);
entscheidung4_av_temp=(sum(entscheidung4(:,1,:)+entscheidung4(:,2,
:)))/(N*2);
for lv3=1:5
    entscheidung3_av(lv3) = entscheidung3_av_temp(:,:,lv3);
    entscheidung4_av(lv3) = entscheidung4_av_temp(:,:,lv3);
end
plot(dat1,'kv','linewidth',2);  plot(dat2,'ko','linewidth',2)
plot(entscheidung3_av,'kv-','linewidth',1);
plot(entscheidung4_av,'ko-','linewidth',1)
title('Comparison of the individual profits in the mass market
experiments','FontWeight','bold');
plot([0  6], [8 8],'-','Linewidth',3,'color','black');
plot([6  6], [0 8],'-','Linewidth',1,'color','black');
text(0.4,4,sprintf('Case 1: 1 round -\n no negotiation'));
text(0.4,7.3,sprintf('Case 2: 1 round -\n negotiation possible'));
text(3,4.5,sprintf('Case 3: 5 rounds -\n no negotiations'));
text(3,7.4,sprintf('Case 4: 5 rounds -\n negotiations possible'));
axis([0 6 3 8]); %grid;
set(gca, 'xtick', 1:1:5);  set(gca, 'ytick', 3:1:8);
xlabel('Round');  ylabel(sprintf('Average individual \n profit per
round'));
```

### 1.4. File: experiment_mass_market_result_bar_chart_bilateral.m

```
clear; scrsz = get(0,'ScreenSize'); hold on;
%vergleiche=[10.13 88.47; 7.53 65.46; 12.62 80.81; 15.91 79.14];

xbeschrift={'(5): PE'; ''; '(5.5)'; '(6)'; '(6.6)'; ''; '(5.5)';
            '(6)'; '(6.6)'; '';  '(6)'; '(6.6)'; ''; '(6.6): NE'}
%kombis=[ 8 0 9 7 4 0 2 6 1 0 2 1 0 8]
kombis=[16 0 3 6 7 0 1 2 4 0 2 2 0 5]
%kombis=[7      0  7    20  37  0   5   20  63 0  8  50 0   23];
%kombis=[65  0  5   13  29  0   2   34  31 0  5  44 0   12];

maxy =max(kombis);
colormap gray;
```

```
xlabel('Decisions of one agent in dependence of the other
participant''s decision');
set(gca,'XTick', 1:length(kombis))
set(gca,'XTickLabel',xbeschrift)
plot([   .5 5.5], [maxy+9 maxy+9],'-
','Linewidth',2,'color','black');
plot([   .5  .5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([ 5.5  5.5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([ 2  2], [maxy+9 0],'--','Linewidth',1,'color','black');
plot([ 6.5  9.5], [maxy+9 maxy+9],'-
','Linewidth',2,'color','black');
plot([ 6.5  6.5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([ 9.5  9.5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([10.5 12.5],   [maxy+9 maxy+9],'-
','Linewidth',2,'color','black');
plot([10.5 10.5],   [maxy+9 0],'-','Linewidth',2,'color','black');
plot([ 12.5  12.5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([13.5 14.5], [maxy+9 maxy+9],'-
','Linewidth',2,'color','black');
plot([13.5 13.5], [maxy+9 0],'-','Linewidth',2,'color','black');
plot([14.5 14.5], [maxy+9 0],'-','Linewidth',2,'color','black');
text(2.8,maxy+15,'(5)');    text(7.8,maxy+15,'(5.5)');
text(11.3,maxy+15,'(6)');   text(13.7,maxy+15,'(6.6)');

for lv=1:length(kombis)
    if (kombis(lv)~=0)  text(lv-
0.1,(kombis(lv)+5),num2str(kombis(lv))); end
end

bar(kombis);
ylim([0 maxy+20]);  %grid
```

### 1.5 File: experiment_mass_market_result_cake_chart_unilateral.m

```
clear; %scrsz = get(0,'ScreenSize'); %hold on;

%static equilibrium
%kombis=[36 20 18 22];      %prozente=[37.5 20.83 18.75 22.92];
%static negotiations
kombis=[48 50 14 18];       prozente=[50 11.5 14.6 23.7];
%dynamic equilibrium
%kombis=[78 100 106 196];   %prozente=[16.3 20.8 22 40.9];
%dynamic negotiation
%kombis=[177 74 101 128];   %prozente=[36.9 15.4 21 26.7];

explode=[1 0 0 0];          h=pie3(kombis,explode);
colormap gray;
textObjs = findobj(h,'Type','text');
pos = get(textObjs, {'Position'});
Names = {'5 units: ';'5.5 units: ';'6 units: '; '6.6 units: '};
```

```
for lv=1:4
    Names(lv) = strcat(Names(lv), num2str(kombis(lv)), ' /
',num2str(prozente(lv)),'%' );
end
newpos=pos;

%dynamic equi
%newpos{1}=newpos{1}+[.3 .3 0]; %newpos{2}=newpos{2}+[-0.1 -0.1
0];
%newpos{4}=newpos{4}+[0.1 0.2 0];
%dynamic negotiation
newpos{1}=newpos{1}+[0.6 0 -1] %newpos{2}=newpos{2}+[-0.1 -0.1 0]
newpos{3}=newpos{3}+[0 0 -0.7]

set(textObjs,{'String'},Names); set(textObjs,
{'Position'},newpos);
```

### 1.6. File: experiment_individual_parameters_rnpvs_expected_events.m

```
clear; scrsz = get(0,'ScreenSize'); figure('Position',[50 100 750
200]);
hold on; T=4; reihe=1; out_x=1:T; out_x0=horzcat(zeros(1,1),out_x)
%-------------------------------------------------------------
%low, complete:    %out_y1=  [50 55 70 85 ];  out_y2=  [45 45 70
95 ];
%low, incomplete:   %out_y1=  [60 50 40 30 ];   out_y2=  [40 45 55
65 ];
%high, complete:    %out_y1=  [80 70 60 50];    out_y2=  [90 75 60
45];
%high, incomplete:
out_y1=  [80 60 50 55];    out_y2=  [70 0 70 70];
%examples chapter 3.6
%out_y1=  [80 80 80 80];  out_y2=  [40 40 40 40];
%out_y1=  [80 70 65 60];  out_y2=  [40 50 70 80];
maxy=max(max(out_y1),max(out_y2));
%-------------------------------------------------------------
matsum1=repmat(out_y1',1,T);          matsum2=repmat(out_y2',1,T);
matsum1=tril(matsum1);                  matsum2=tril(matsum2);
rnpv1=sum(matsum1);                     rnpv2=sum(matsum2);
rnpv1=horzcat(rnpv1,zeros(1,1));
rnpv2=horzcat(rnpv2,zeros(1,1));
max_rnpv=max(max(rnpv1),max(rnpv2));
%-------------------------------------------------------------------
%1.row
subplot(1,3,1); plot(out_x,out_y1,'kx-','Linewidth',2);     grid
on;
text(out_x(2),out_y1(2)+5,'PV''s of S1');
ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+50]); set(gca, 'xtick',
min(out_x):1:max(out_x));
```

```
subplot(1,3,2); plot(out_x,out_y2,'kx-','Linewidth',1);     grid
on;
text(out_x(2),out_y2(2)-5,'PV''s of S2');
ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+50]);  set(gca, 'xtick',
min(out_x):1:max(out_x));

matsum1=repmat(out_y1',1,T);    matsum2=repmat(out_y2',1,T);
matsum1=tril(matsum1);          matsum2=tril(matsum2);
rnpv1=sum(matsum1);             rnpv2=sum(matsum2);
rnpv1=horzcat(rnpv1,zeros(1,1));rnpv2=horzcat(rnpv2,zeros(1,1));

subplot(1,3,3);
plot(out_x0,rnpv1,'kx-','Linewidth',2); hold on;
text(out_x0(2),rnpv1(2)-10,'residual NPV''s of S1');
plot(out_x0,rnpv2,'kx-','Linewidth',1);
text(out_x0(2),rnpv2(2)+10,'residual NPV''s of S2');
%grid on;
axis([0 (T+1) 0 max_rnpv+20])
set(gca, 'xtick', min(out_x):1:max(out_x));
```

### 1.7. File: experiment_individual_parameters_rnpvs_unexpected_events.m

```
clear; scrsz = get(0,'ScreenSize'); figure('Position',[50 100 750
400]);
hold on;
T=4; reihe=1; out_x=1:T; yoben = 20;
out_x0=horzcat(zeros(1,1),out_x)
%out_y1=ones(1,T)*11; %constant
%out_y2=ones(1,T)*6; %constant
%out_y1=[2 3 16 5 7 3 8 9 7 13 9 6 5 3 6 14 18 13 10 9];  %no
crossing
%out_y2=[4 3 5 5 6 9 8 5 13 6 8 16 7 6 4 5 2 8 8 4]; %no crossing
%-------------------------------------------------------------
% example chapter 3.6.5
shock1=2; shock2=3;
out_y1=  [ 30  40  55 65];
outy1_u1=[  0  40  55 65];
outy1_u2=[  0   0  75 85]; %change shock 2
out_y2=  [45 30 40 55];
outy2_u1=[0  60 65 70];    %change shock 1
outy2_u2=[0   0 65 70];

% only public information
shock1=2; shock2=3;
out_y1=  [ 40  45  55 65];
outy1_u1=[  0  45  55 65];
outy1_u2=[  0   0  70 80]; %change shock 2
out_y2=  [50 45 50 55];
outy2_u1=[0  60 65 70];    %change shock 1
```

```
outy2_u2=[0   0 65 70];

% also private information
%shock1=2; shock2=3;
%out_y1=  [ 65  55  50 40];  outy1_u1=[  0  45  40 35]; %change
shock 1
%outy1_u2=[  0   0  40 35];
%out_y2=  [55 45 35 30];
%outy2_u1=[0  45 35 30];     outy2_u2=[0   0 45 50];  %change
shock 2

maxy=max(max(out_y1),max(out_y2));
%-----------------------------------------------------------------
matsum1=repmat(out_y1',1,T);          matsum2=repmat(out_y2',1,T);
matsum1_u1=repmat(outy1_u1',1,T);
matsum2_u1=repmat(outy2_u1',1,T);
matsum1_u2=repmat(outy1_u2',1,T);
matsum2_u2=repmat(outy2_u2',1,T);
matsum1=tril(matsum1);                 matsum2=tril(matsum2);
matsum1_u1=tril(matsum1_u1);
matsum2_u1=tril(matsum2_u1);
matsum1_u2=tril(matsum1_u2);          matsum2_u2=tril(matsum2_u2);
rnpv1=sum(matsum1);                     rnpv2=sum(matsum2);
rnpv1_u1=sum(matsum1_u1);               rnpv2_u1=sum(matsum2_u1);
rnpv1_u2=sum(matsum1_u2);               rnpv2_u2=sum(matsum2_u2);
rnpv1=horzcat(rnpv1,zeros(1,1));
rnpv2=horzcat(rnpv2,zeros(1,1));
rnpv1_u1=horzcat(rnpv1_u1,zeros(1,1));
rnpv2_u1=horzcat(rnpv2_u1,zeros(1,1));
rnpv1_u2=horzcat(rnpv1_u2,zeros(1,1));
rnpv2_u2=horzcat(rnpv2_u2,zeros(1,1));
max_rnpv=max(max(rnpv1),max(rnpv2));
%-----------------------------------------------------------------
%1.row
subplot(3,3,1); plot(out_x,out_y1,'kx-','Linewidth',2), grid
axis([0 (T+1) 0 maxy+yoben])
set(gca, 'xtick', min(out_x):1:max(out_x));
subplot(3,3,2); plot(out_x,out_y2,'kx-','Linewidth',1), grid
ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+yoben])
set(gca, 'xtick', min(out_x):1:max(out_x));

matsum1=repmat(out_y1',1,T);    matsum2=repmat(out_y2',1,T);
matsum1=tril(matsum1);          matsum2=tril(matsum2);
rnpv1=sum(matsum1);             rnpv2=sum(matsum2);
rnpv1=horzcat(rnpv1,zeros(1,1));rnpv2=horzcat(rnpv2,zeros(1,1));
subplot(3,3,3);
plot(out_x0,rnpv1,'k-','Linewidth',2); hold on;
plot(out_x0,rnpv2,'k-','Linewidth',1);
%grid on;
axis([0 (T+1) 0 max_rnpv+yoben])
```

```
set(gca, 'xtick', min(out_x):1:max(out_x));
%-----------------------------------------------------------------
%2.row
subplot(3,3,4);
plot(out_x(shock1:T),out_y1(shock1:T),'kx:','Linewidth',1); hold
on;
plot(out_x(shock1:T),outy1_u1(shock1:T),'kx-','Linewidth',2); hold
on;
plot([shock1 shock1],[0 maxy+yoben],'-
','Linewidth',2,'color','black');
grid on;
axis([0 (T+1) 0 maxy+yoben]);
set(gca, 'xtick', min(out_x):1:max(out_x));
subplot(3,3,5);
plot(out_x(shock1:T),out_y2(shock1:T),'kx:','Linewidth',1); hold
on;
plot(out_x(shock1:T),outy2_u1(shock1:T),'kx-','Linewidth',1); hold
on;
plot([shock1 shock1],[0 maxy+yoben],'-
','Linewidth',2,'color','black');
grid on;    ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+yoben]); set(gca, 'xtick',
min(out_x):1:max(out_x));
subplot(3,3,6);
plot(out_x0(shock1+1:T+1),rnpv1(shock1+1:T+1),'k:','Linewidth',1);
hold on;
plot(out_x0(shock1+1:T+1),rnpv2(shock1+1:T+1),'k:','Linewidth',1);
hold on;
plot(out_x0(shock1+1:T+1),rnpv1_u1(shock1+1:T+1),'k-
','Linewidth',2); hold on;
plot(out_x0(shock1+1:T+1),rnpv2_u1(shock1+1:T+1),'k-
','Linewidth',1); hold on;

plot([shock1 shock1],[0 max_rnpv+yoben],'-
','Linewidth',2,'color','black');
%grid on;
axis([0 (T+1) 0 max_rnpv+yoben]);
set(gca, 'xtick', min(out_x):1:max(out_x));
%-----------------------------------------------------------------
%3.row
subplot(3,3,7);
plot(out_x(shock2:T),outy1_u1(shock2:T),'kx:','Linewidth',2); hold
on;
plot(out_x(shock2:T),outy1_u2(shock2:T),'kx-','Linewidth',2); hold
on;
plot([shock2 shock2],[0 maxy+yoben],'-
','Linewidth',2,'color','black');
grid on;
ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+yoben])
set(gca, 'xtick', min(out_x):1:max(out_x));
```

```
subplot(3,3,8);
plot(out_x(shock2:T),outy2_u1(shock2:T),'kx:','Linewidth',1); hold
on;
plot(out_x(shock2:T),outy2_u2(shock2:T),'kx-','Linewidth',1); hold
on;
plot([shock2 shock2],[0 maxy+yoben],'-
','Linewidth',2,'color','black');
grid; ylim([0 maxy+5]);
axis([0 (T+1) 0 maxy+yoben]); set(gca, 'xtick',
min(out_x):1:max(out_x));
subplot(3,3,9);
plot(out_x0(shock2+1:T+1),rnpv1_u1(shock2+1:T+1),'kx:','Linewidth'
,1); hold on;
plot(out_x0(shock2+1:T+1),rnpv2_u1(shock2+1:T+1),'kx:','Linewidth'
,1); hold on;
plot(out_x0(shock2+1:T+1),rnpv1_u2(shock2+1:T+1),'kx-
','Linewidth',2); hold on;
plot(out_x0(shock2+1:T+1),rnpv2_u2(shock2+1:T+1),'kx-
','Linewidth',1); hold on;
plot([shock2 shock2],[0 max_rnpv+yoben],'-
','Linewidth',2,'color','black');
%grid on;
axis([0 (T+1) 0 max_rnpv+yoben]);
set(gca, 'xtick', min(out_x):1:max(out_x));
```

### 1.8. File: experiment_individual_result_calculations_table.m

```
clear;
verzeichnis='c:\game theory\ergebnisse\phase 2010-2
dissertation\MS Excell experimental data\';
datei ='collected data ind-dynamic - expected - gefiltert.xls';
pfad = [verzeichnis, datei];
% [40 161; 32 129; 34 137; 34 137 ;;; 39 157; 33 133]
%OOCS = [47.5  2.5 0; 50    5   0  ; 70    0        0 ; 90   0
5  ];  %e,low
%AD   = [25   25   0; 27.5 27.5 0  ; 35   17.5   17.5 ; 42.5 0
42.5];  %e,low
    %N=40; dat1 = xlsread(pfad, 1, 'B3:E161');
%OOCS = [50   10   0; 47.5 2.5  0  ; 47.5  0      7.5 ; 47.5 0
17.5];  %e,low
%AD   = [30   30   0; 25   25   0  ; 27.5  0     27.5 ; 32.5 0
32.5];  %e,low
    %N=32; dat1 = xlsread(pfad, 2, 'B3:E129');
%OOCS = [85    0   5;  0    0   0  ;  0    0   0 ;  0   0     0
];  %high
%AD   = [45    0  45; 37.5  0  37.5; 30   15   15 ; 22.5 0
22.5]; %high
%    N=34; dat1 = xlsread(pfad, 3, 'B3:E137');
OOCS = [75   10   0;  0    0   0  ;  0    0        0 ; 0   0     0
];  %high
```

```
AD   = [40   40   0; 30   30   0  ; 25   25        0 ; 27.5 27.5
0  ]; %high
    N=34; dat1 = xlsread(pfad, 4, 'B3:E137');
%datei ='collected data ind-dynamic - unexpected - gefiltert.xls';
%OOCS = [45    0   5; 52.5  0  7.5 ; 67.5 2.5       0 ; 75   5
0  ]; %u,low
%AD   = [25    0  25; 30    0  30  ; 35   35        0 ; 40   40
0  ]; %u,low
    %N=40; dat1 = xlsread(pfad, 1, 'B3:E161');
%OOCS = [60    5   0; 45    0   0  ; 42.5 0       2.5 ; 42.5 0
7.5]; %u,low
%AD   = [32.5 32.5 0; 22.5 22.5 0  ; 22.5 0      22.5 ; 25   0
25 ]; %u,low
    %N=33; dat1 = xlsread(pfad, 2, 'B3:E133');

OOCS = transpose(OOCS);  AD = transpose(AD);
OOCS_hori = OOCS(:)'; AD_hori   = AD(:)'; P = []; S1 = []; S2 =
[];
for lv=1:N
  P =  [P; dat1(((lv-1)*4+3),:)];
  S1 = [S1; dat1(((lv-1)*4+1),:)]; S2 = [S2; dat1(((lv-
1)*4+2),:)];
end
P_av  = mean(P); S1_av = mean(S1);  S2_av = mean(S2);
ratdat_av=[];
for lv2=1:4
   ratdat_av=[ratdat_av P_av(lv2) S1_av(lv2) S2_av(lv2)];
end

v_AD_matrix   = [P_av; S1_av; S2_av]' ./ AD';
v_OOCS_matrix = [P_av; S1_av; S2_av]' ./ OOCS';
v_AD   = ratdat_av ./ AD_hori; v_AD = v_AD(find(~isnan(v_AD) &
~isinf(v_AD)));
AD_einsen=ones(1,length(v_AD));
v_AD_betrag = abs(v_AD - AD_einsen);
v_AD_betrag_av = mean(v_AD_betrag');
v_OOCS   = ratdat_av ./ OOCS_hori;
v_OOCS = v_OOCS(find(~isnan(v_OOCS) & ~isinf(v_OOCS)));
OOCS_einsen=ones(1,length(v_OOCS));
v_OOCS_betrag = abs(v_OOCS - OOCS_einsen);
v_OOCS_betrag_av = mean(v_OOCS_betrag');

% high switching costs
v_OOCS_betrag_av_round1 = mean(v_OOCS_betrag);
v_AD_betrag_av_round1   = (v_AD_betrag(1)+v_AD_betrag(2))/2;
%v_AD_betrag_av_round234 =
(v_AD_betrag(3)+v_AD_betrag(4)+v_AD_betrag(5)+v_AD_betrag(7)+v_AD_
betrag(8)+v_AD_betrag(9))/6;
v_AD_betrag_av_round234 =
(v_AD_betrag(3)+v_AD_betrag(4)+v_AD_betrag(5)+v_AD_betrag(6)+v_AD_
betrag(7)+v_AD_betrag(8))/6;
```

### 1.9. File: experiment_individual_result_low_costs_figure.m

```
clear; scrsz = get(0,'ScreenSize'); figure('Position',[50 100 800
250]); hold on;

result= [10.13 88.47; 7.53 65.46; 12.62 80.81; 15.91 79.14];
links1=0.6; links2=2.45;  rechts1=2.55; rechts2=4.4; hoehe= 130;
colormap(gray);
plot([ links1  links2], [hoehe hoehe],'-
','Linewidth',2,'color','black');
plot([ links1  links1], [hoehe-10 hoehe],'-
','Linewidth',2,'color','black');
plot([ links2  links2], [hoehe-10 hoehe],'-
','Linewidth',2,'color','black');
plot([ rechts1  rechts2], [hoehe hoehe],'-
','Linewidth',2,'color','black');
plot([ rechts1  rechts1], [hoehe-10 hoehe],'-
','Linewidth',2,'color','black');
plot([ rechts2  rechts2], [hoehe-10 hoehe],'-
','Linewidth',2,'color','black');
text(1-0.3,25,strcat(num2str(result(1,1)), '%'));
text(2-0.3,25,strcat(num2str(result(2,1)), '%'));
text(3-0.3,25,strcat(num2str(result(3,1)), '%'));
text(4-0.3,25,strcat(num2str(result(4,1)), '%'));
text(1,100,strcat(num2str(result(1,2)), '%'));
text(2,100,strcat(num2str(result(2,2)), '%'));
text(3,100,strcat(num2str(result(3,2)), '%'));
text(4,100,strcat(num2str(result(4,2)), '%'));
text(1.1,hoehe+10,'Only expected events');
text(3.1,hoehe+10,'Also unexpected events');
text(0.65,hoehe-10,'Only public information');  text(1.55,hoehe-
10,'Also private information');
text(2.6,hoehe-10,'Only public information');   text(3.5,hoehe-
10,'Also private information');
title(sprintf('Experiments with low agent switching costs: \n
Comparison between the OOCS-rule and the AD-rule \n regarding the
average deviations (absolute values) between \n the theoretic
predictions and the experimental data'),'FontWeight','bold');
xbeschrift = {'OOCS     AD  '; 'OOCS     AD  '; 'OOCS     AD  ';
'OOCS     AD  '};
ybeschrift = {'0'; '50%'; '100%'; ''};
set(gca,'XTick', 1:4);
set(gca,'XTickLabel',xbeschrift);
set(gca,'YTickLabel',ybeschrift);
ylim([0 150]);                          bar(result,0.9 );
```

## 2.1. Files: 4-2-2-1-cournot-equilibrium.ztt and 4-2-2-2-cournot-negotiations.ztt
## Excerpts with source codes:

**Background:**

```
subjects.do (
        // Default payoffs for 2 agents and 4 alternatives for both
        TotalProfit = 1000;
        array pbar[2];
        Pi11 = 1 ; Pi12 = 1 ; Pi13 = 1; Pi14 = 1;
        Pi21 = 1 ; Pi22 = 1 ; Pi23 = 1; Pi24 = 1;
        Pi31 = 1 ; Pi32 = 1 ; Pi33 = 1; Pi34 = 1;
        Pi41 = 1 ; Pi42 = 1 ; Pi43 = 1; Pi44 = 1;
)
```

**Stage "Decision":**

```
        // this player prefers: 1
        Pi11 = 8;    Pi12 = 5;   Pi13 = 3;   Pi14 = 1;
        Pi21 = 10;Pi22 = 7;      Pi23 = 4;   Pi24 = 2;
        Pi31 = 11;Pi32 = 8;      Pi33 = 5;   Pi34 = 3;
        Pi41 = 12;Pi42 = 9; Pi43 = 6;        Pi44 = 4;
```

**Stage "Profit display":**

```
// Pnum1 = NumPeriods;
// array ProfitHistory[20];
// array ProfitOthersHistory[20];

OthersChoice = find (same(Group) & not ( same (Subject)), Choice );
OthersTotalProfit = find (same(Group) & not ( same (Subject)), TotalProfit );

OthersPi11 =find (same(Group) & not ( same (Subject)), Pi11);
OthersPi12 =find (same(Group) & not ( same (Subject)), Pi12);
OthersPi13 =find (same(Group) & not ( same (Subject)), Pi13);
OthersPi14 =find (same(Group) & not ( same (Subject)), Pi14);
OthersPi21 =find (same(Group) & not ( same (Subject)), Pi21);
OthersPi22 =find (same(Group) & not ( same (Subject)), Pi22);
OthersPi23 =find (same(Group) & not ( same (Subject)), Pi23);
OthersPi24 =find (same(Group) & not ( same (Subject)), Pi24);
OthersPi31 =find (same(Group) & not ( same (Subject)), Pi31);
OthersPi32 =find (same(Group) & not ( same (Subject)), Pi32);
OthersPi33 =find (same(Group) & not ( same (Subject)), Pi33);
OthersPi34 =find (same(Group) & not ( same (Subject)), Pi34);
OthersPi41 =find (same(Group) & not ( same (Subject)), Pi41);
OthersPi42 =find (same(Group) & not ( same (Subject)), Pi42);
OthersPi43 =find (same(Group) & not ( same (Subject)), Pi43);
OthersPi44 =find (same(Group) & not ( same (Subject)), Pi44);
if ( Choice == 1) { if (OthersChoice ==1)  { Profit=Pi11;   OthersProfit=OthersPi11;  } };
if ( Choice == 1) { if (OthersChoice ==2)  { Profit=Pi12;   OthersProfit=OthersPi21;  } };
if ( Choice == 1) { if (OthersChoice ==3)  { Profit=Pi13;   OthersProfit=OthersPi31;  } };
```

```
if ( Choice == 1) { if (OthersChoice ==4)  { Profit=Pi14;   OthersProfit=OthersPi41;  } };
if ( Choice == 2) { if (OthersChoice ==1)  { Profit=Pi21;   OthersProfit=OthersPi12;  } };
if ( Choice == 2) { if (OthersChoice ==2)  { Profit=Pi22;   OthersProfit=OthersPi22;  } };
if ( Choice == 2) { if (OthersChoice ==3)  { Profit=Pi23;   OthersProfit=OthersPi32;  } };
if ( Choice == 2) { if (OthersChoice ==4)  { Profit=Pi24;   OthersProfit=OthersPi42;  } };
if ( Choice == 3) { if (OthersChoice ==1)  { Profit=Pi31;   OthersProfit=OthersPi13;  } };
if ( Choice == 3) { if (OthersChoice ==2)  { Profit=Pi32;   OthersProfit=OthersPi23;  } };
if ( Choice == 3) { if (OthersChoice ==3)  { Profit=Pi33;   OthersProfit=OthersPi33;  } };
if ( Choice == 3) { if (OthersChoice ==4)  { Profit=Pi34;   OthersProfit=OthersPi43;  } };
if ( Choice == 4) { if (OthersChoice ==1)  { Profit=Pi41;   OthersProfit=OthersPi14;  } };
if ( Choice == 4) { if (OthersChoice ==2)  { Profit=Pi42;   OthersProfit=OthersPi24;  } };
if ( Choice == 4) { if (OthersChoice ==3)  { Profit=Pi43;   OthersProfit=OthersPi34;  } };
if ( Choice == 4) { if (OthersChoice ==4)  { Profit=Pi44;   OthersProfit=OthersPi44;  } };

if ((Subject==1) & (Period==:Period) )
{
  pastprofits.new
  {
    incomeS1=-10; incomeS2=-10;
  }
}
pastprofits.do
{
 if ((:Subject==1) & (Period==:Period) ) { incomeS1=:Profit; }
 if ((:Subject==2) & (Period==:Period) ) { incomeS2=:Profit; }}
```

## 2.3. Files: 5-3-2-1 game dynamic low expected complete.ztt, 5-3-2-2 game dynamic low expected incomplete.ztt, 5-3-3-1 game dynamic low unexpected complete.ztt, 5-3-3-2 game dynamic low unexpected incomplete.ztt

## Excerpts with source codes:

**Background:**

```
globals.do (
SELLERTYPE = 1;
BUYERTYPE=2;
TimeAuction = 90;
TradeCounter = 0;
MaxTrades = 1;
Schluesselmarker = 101;
)
subjects.do (
Type = 0;
RValue = 0; RValueOS = 0;
MaxP = 1000; MinP = 0;
NumTrades = 0 ;
OthersProfit = 0; OthersTotalProfit = 0;
```

```
pwahl1 = 0;
if (Period >1 ) { pwahl1 = OLDsubjects.find( same( Subject ) , pwahl1 ); }
TreatmentDone = 0;
if (Period >1 ) { TreatmentDone = OLDsubjects.find( same( Subject ) , TreatmentDone ); }

profitscosts.new  {   array RValue1[20]; array RValue2[20]; }
profitscosts.do
{
  RValue1[1] = 50;   RValue1[2] = 55;   RValue1[3] = 70;   RValue1[4] = 85;
  RValue2[1] = 45;   RValue2[2] = 45;   RValue2[3] = 70;   RValue2[4] = 95;
}
)
contracts.do (
Buyer = -1; Seller = -1; Price = -1; Creator = -1; TradeID =-1;
)
partner.do (
partnerwahl = 0;
if (Period >1 ) {
  partnerwahl = OLDsubjects.find( partnerwahl );
}
)
```

**Stage „Seller Auction“:**

```
subjects.do (
boolvar1 = 0;
if ( ((pwahl1==0) | (pwahl1==Subject)) & (TreatmentDone == 0)    )   {boolvar1=1; }
Participate = if ((Type == SELLERTYPE) & (boolvar1==1) ,1,0);
if (Subject==1)
{
RValue = profitscosts.find(  same(Group), RValue1[:Period]);
            RValueOS = profitscosts.find(  same(Group), RValue2[:Period]);
}
if (Subject==2)
{
             RValue = profitscosts.find(  same(Group), RValue2[:Period]);
             RValueOS = profitscosts.find(  same(Group), RValue1[:Period]);
}
)

Active screen → Container → Button-sell → contracts.do (
Seller = : Subject; TradeID = TradeCounter;
subjects.do {
if ( Subject == : Buyer ) {  NumTrades = NumTrades +1;  Profit = Price ; }
if ( Subject == : Seller ) {  NumTrades = NumTrades +1;  Profit = RValue - Price ;  }
}
contracts.do {
             if ( Seller == :Seller & Buyer == -1 ) {  Buyer = -2;  }
             if ( Buyer == :Buyer & Seller == -1 ) {  Seller = -2;  }
}
if  (NumTrades == 1)  {  subjects.do { if (  same(Group)  )  {    LeaveStage=1;   }  }   }
```

```
)
```

**Stage „Buyer Auction“:**

```
subjects.do (
Participate = if (  (Type == BUYERTYPE)  ,1,0);
RValue = profitscosts.find(  same(Group), RValue1[:Period]);
RValueOS = profitscosts.find(  same(Group), RValue2[:Period]);
RValueMax = max ( RValue, RValueOS );
)
Active screen → Container → Button-buy → contracts.do (
Buyer = : Subject; TradeID = TradeCounter;
subjects.do {
if ( Subject == : Buyer ) {  NumTrades = NumTrades +1;  Profit = Price ; }
if ( Subject == : Seller ) {  NumTrades = NumTrades +1;  Profit = RValue - Price ;  }
}
contracts.do{
  if( Seller == :Seller & Buyer == -1 ) {  Buyer = -2; }
  if( Buyer == :Buyer & Seller == -1 ) {  Seller = -2; }
}
if  (NumTrades == 1)  {  subjects.do { if (  same(Group)  )  {    LeaveStage=1;   }  }   }
)
```

**Stage „Profit display“:**

```
subjects.do (
ProfitS1 = find (same(Group) & (Subject==1), Profit);
ProfitS2 = find (same(Group) & (Subject==2), Profit);
ProfitP =   find (same(Group) & (Subject==3), Profit);
TotalProfitS1 = find (same(Group) & (Subject==1), TotalProfit);
TotalProfitS2 = find (same(Group) & (Subject==2), TotalProfit);
TotalProfitP =   find (same(Group) & (Subject==3), TotalProfit);
if ((Subject==1) & (Period==:Period) )
{
   pastprofits.new
   {
     incomeS1=-10; incomeS2=-10;  incomeP=-10;
   }
}
pastprofits.do
{
  if ((:Subject==1) & (Period==:Period) ) { incomeS1=:Profit; }
  if ((:Subject==2) & (Period==:Period) ) { incomeS2=:Profit; }
  if ((:Subject==3) & (Period==:Period) ) { incomeP=:Profit; }
}
)
```

## 2.5. Files: 5-3-2-3 game dynamic high expected complete.ztt, 5-3-2-4 game dynamic high expected incomplete.ztt
## Excerpts with source codes:

**Stage „Seller Auction“:**

```
subjects.do (
boolvar1 = 0;
if ( ((pwahl1==0) | (pwahl1==Subject)) & (TreatmentDone == 0)    )   {boolvar1=1; }

Participate = if ((Type == SELLERTYPE) & (boolvar1==1) ,1,0);
if (Subject==1)
{
  RValue = profitscosts.find(  same(Group), RValue1[:Period]);
//   RNPV =   profitscosts.find(  same(Group), RNPV1[:Period]);
  RValueOS = profitscosts.find(  same(Group), RValue2[:Period]);
 //  RNPVOS =   profitscosts.find(  same(Group), RNPV2[:Period]);
}

if (Subject==2)
{
    RValue = profitscosts.find(  same(Group), RValue2[:Period]);
//    RNPV =   profitscosts.find(  same(Group), RNPV2[:Period]);
    RValueOS = profitscosts.find(  same(Group), RValue1[:Period]);
 //   RNPVOS =        profitscosts.find(  same(Group), RNPV1[:Period]);
}
)

Active screen → Container (Negotiation) → Button-sell → contracts.do (
Seller = : Subject; TradeID = TradeCounter;
if ((pwahl1 != 1) & (pwahl1 != 2))  {
   partner.new  { partnerwahl = :Subject;  }
   partner.do {
        subjects.do {   pwahl1 = :partnerwahl;  }
   }
}
subjects.do {
if ( Subject == : Buyer ) {  NumTrades = NumTrades +1;  Profit = Price ; }
if ( Subject == : Seller ) {  NumTrades = NumTrades +1;  Profit = RValue - Price ;  }
}
contracts.do {
  if ( Seller == :Seller & Buyer == -1 ) {  Buyer = -2;  }
  if ( Buyer == :Buyer & Seller == -1 ) {  Seller = -2;  }
}
if  (NumTrades == 1)  {  subjects.do { if (  same(Group)  )  {    LeaveStage=1;   }  }   }
)

Active screen → Container (Renegotiation) → Button-sell → contracts.do (
Seller = : Subject; TradeID = TradeCounter;
```

```
subjects.do {
if ( Subject == : Buyer ) {  NumTrades = NumTrades +1;  Profit = Price ; }
if ( Subject == : Seller ) {  NumTrades = NumTrades +1;  Profit = RValue - Price ;  }
}
contracts.do {
  if ( Seller == :Seller & Buyer == -1 ) {  Buyer = -2;  }
  if ( Buyer == :Buyer & Seller == -1 ) {  Seller = -2;  }
}
if  (NumTrades == 1)  {  subjects.do { if (  same(Group)  )  {    LeaveStage=1;   }  }   }
)
```

**Stage „Buyer Auction“:**

```
subjects.do (
if (TreatmentDone == 0)   {boolvar1=1; }
Participate = if (  (Type == BUYERTYPE) & (boolvar1==1)  ,1,0);
RValue = profitscosts.find(  same(Group), RValue1[:Period]);
// RNPV =   profitscosts.find(  same(Group), RNPV1[:Period]);
RValueOS = profitscosts.find(  same(Group), RValue2[:Period]);
// RNPVOS =   profitscosts.find(  same(Group), RNPV2[:Period]);
RValueMax = max ( RValue, RValueOS );
)
Active screen → Container (Renegotiation) → Button-buy → contracts.do (
Buyer = : Subject; TradeID = TradeCounter;
subjects.do {
if ( Subject == : Buyer ) {  NumTrades = NumTrades +1;  Profit = Price ; }
if ( Subject == : Seller ) {  NumTrades = NumTrades +1;  Profit = RValue - Price ;  }
}
contracts.do{
  if( Seller == :Seller & Buyer == -1 ) {  Buyer = -2; }
  if( Buyer == :Buyer & Seller == -1 ) {  Seller = -2; }
}
if  (NumTrades == 1)  {  subjects.do { if (  same(Group)  )  {    LeaveStage=1;   }  }   }
)
```