

IIR Filter design

- 1
- 2 Analog Filters
- 3 Projektowanie filtrów cyfrowych Butterwortha i Czebyszewa
- 4 Projekt filtru dolnoprzepustowego
- 5 Reakcja filtru Butterworth na częstotliwości odcięcia
- 6 Analiza filtrów Butterwortha w Mathematica
- 7 Analiza względem wszystkich parametrów
- 8 Zależność błędu od rzędu filtra
- 9 Nyquist Ploty analogowych
- 10 Bode plots
- 11 Inna ilość bitów
- 12 Analiza współczynników filtrów
- 13 Filtry cyfrowe
- 14 Badanie stabilności filtrów cyfrowych (6 bit)

14.1 Ustalenie postaci filtrów cyfrowych

Poniżej znajdują się definicje filtrów analogowych i ich cyfrowych odpowiedników w postaci tabel dla rzędu $N=\{1,2,\dots,7\}$ oraz częstotliwości $\omega=\{\omega_{\min}, \omega_{\max}, \Delta\omega\}$

14.1.1 Definicje

```
In[290]:= Δω = 0.3;  
ωmin = 0.1;  
ωmax = 2.0;
```

14.1.2 Butterworth

```
In[293]:= bmodels = Table[TransferFunctionFactor@ButterworthFilterModel[{"Lowpass", n, \omega}, s], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];  
DGbmodels =  
Table[TransferFunctionFactor@  
ToDiscreteTimeModel[ButterworthFilterModel[{"Lowpass", n, \omega}, s], 1], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];
```

14.1.3 Chebyshev 1

```
In[295]:= c1models = Table[TransferFunctionFactor@Chebyshev1FilterModel[{"Lowpass", n, \omega}, s], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];  
DGc1models =  
Table[TransferFunctionFactor@ToDiscreteTimeModel[Chebyshev1FilterModel[{"Lowpass", n, \omega}, s], 1], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];
```

14.1.4 Chebyshev 2

```
In[297]:= c2models = Table[TransferFunctionFactor@Chebyshev2FilterModel[{n, \omega}, s], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];  
DGc2models =  
Table[TransferFunctionFactor@ToDiscreteTimeModel[Chebyshev2FilterModel[{n, \omega}, s], 1], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];
```

14.1.5 Elliptic

```
In[299]:= emodels = Table[TransferFunctionFactor@EllipticFilterModel[{n, \omega}, s], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];  
DGemodels = Table[TransferFunctionFactor@ToDiscreteTimeModel[EllipticFilterModel[{n, \omega}, s], 1], {n, 1, 7, 1}, {\omega, \omega_{min}, \omega_{max}, \Delta\omega}];
```

14.2 Dyskretyzacja na poziomie zer i biegunów

14.2.1 Wstęp

Na początek przedstawiam sposób w jaki dyskretyzowałem dotychczas, tj. mając filtr cyfrowy w postaci

$$\frac{a_0(z - a_1)}{(z - b_1)(z - b_2)} \text{ dyskretyzowałem liczby } a_1, b_1, b_2.$$

Okazuje się, że nie jest najlepszy sposób dyskretyzacji by sprawdzić rzeczywiste efekty tego procesu.

Lepszy sposób prezentowany w kolejnym paragrafie, polega na dyskretyzacji końcowych współczynników filtra, a więc współczynników c_0, c_1, d_0, d_1, d_2 filtru $\frac{c_0 + c_1 z}{d_0 + d_1 z + d_2 z^2}$ będącego wymnożoną postacią wyżej przedstawionego filtru.

14.2.2 Definicje

```
In[3]:= DiscretizeList[x_, max_, bits_] :=
  If[# ≥ 0, 1, -1] Round[(Abs[#]/max) (Power[2, bits - 1] - 1)] max / (Power[2, bits - 1] - 1) & /@ x;

In[279]:= DiscretizeComplexLists[x_, y_, coeff_, bits_] := Module[{xD, yD, coeffD, max},
  If[y == {}, {}, max = Max[Abs[Join[Re@x, Im@x, Re@y, Im@y, Re@{coeff}, Im@{coeff}]]];
  xD = DiscretizeList[Re@x, max, bits] + i DiscretizeList[Im@x, max, bits];
  yD = DiscretizeList[Re@y, max, bits] + i DiscretizeList[Im@y, max, bits];
  coeffD = Total[DiscretizeList[{Re@coeff, i Im@coeff}, max, bits]];
  {xD, yD, coeffD}];
];

DiscretizeModel[tf_, bits_] := Module[{tfD, coeff, zeros, poles, coeffD, zerosD, polesD},
  coeff = First[Flatten[Numerator[tf[z]]]];
  coeff = If[Length[coeff] > 0, coeff[[1]], coeff];
  zeros = (z /. Solve[Numerator@(tf[z]) == 0, z]) /. {z → {}};
  poles = (z /. Solve[Denominator@(tf[z] / coeff) == 0, z]);
  {zerosD, polesD, coeffD} = DiscretizeComplexLists[zeros, poles, Last@poles, bits];
  (*nie wykorzystuję coeffD, bo często jest bardzo mały i przechodzi w 0 podczas dyskretyzacji, by nie zaburzyć wyniku, podaję mu inny, istniejący biegum*)
  Chop[coeff (z - zerosD /. {List → Times}) / (z - polesD /. {List → Times})]
];
];
```

Pomocnicza funkcja by pobrać biegumy zdyskretyzowanych filtrów

```
In[53]:= ExtractPoles[tf_] := Module[{coeff},
  coeff = First[Flatten[Numerator[tf]]];
  coeff = If[Length[coeff] > 0, coeff[[1]], coeff];
  (z /. Solve[Denominator@(tf / coeff) == 0, z])
];
```

Ustalamy ilość bitów do symulacji

```
In[365]:= bity = 6;
```

14.2.3 Testy

```
In[54]:= {a, b, c} = DiscretizeComplexLists[{1, 2, 3}, {i, i 2, i 3}, 2, 6]
Out[54]= {30/31, 63/31, 3}, {30 i/31, 63 i/31, 3 i}, 63/31}
```

Dyskretyzujemy na poziomie filtrów cyfrowych

```
In[13]:= buttModel = DGbmodels[[1, 1]]
```

$$\text{Out[13]} = \left(\frac{0.047619 \left(1 + \frac{i}{z}\right)}{-0.904762 + \frac{i}{z}} \right)_1^{\mathcal{T}}$$

```
In[61]:= polesButt = TransferFunctionPoles[buttModel][[1, 1]]
```

```
Out[61]= {0.904762}
```

```
In[231]:= buttModelD = DiscretizeModel[buttModel, bity]
```

$$\frac{0.047619 (1. + z)}{-0.903226 + z}$$

```
In[60]:= polesButtD = ExtractPoles[buttModelD]
```

```
Out[60]= {0.90411}
```

Ale funkcja dyskretyzująca działa też dla filtrów analogowych

```
In[15]:= buttModel2 = bmodels[[3, 3]]
```

$$\text{Out}[15]= \left(\frac{0.125}{((0.25 - 0.433013 i) + s)((0.25 + 0.433013 i) + s)(0.5 + s)} \right)^{\mathcal{T}}$$

```
In[281]:= buttModel2D = DiscretizeModel[buttModel2, bity]
```

$$\text{Out}[281]= \frac{0.125}{((0.258065 - 0.435484 i) + z)((0.258065 + 0.435484 i) + z)(0.5 + z)}$$

14.2.4 Dyskretyzacja

```
In[301]:= DGbmodelsD = Map[DiscretizeModel[#, bity] &, DGbmodels, {2}];
```

```
In[302]:= DGc1modelsD = Map[DiscretizeModel[#, bity] &, DGc1models, {2}];
```

```
In[303]:= DGc2modelsD = Map[DiscretizeModel[#, bity] &, DGc2models, {2}];
```

```
In[304]:= DGmodelsD = Map[DiscretizeModel[#, bity] &, DGmodels, {2}];
```

14.2.5 Porównanie położenia biegunków

★ Definicje

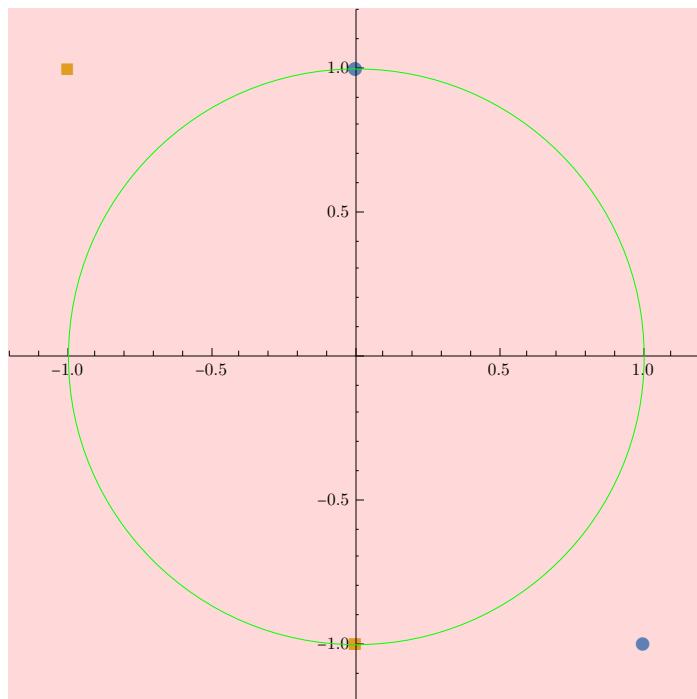
Przygotowuję funkcję rysującą biegunki i zaznaczającą na czerwono te w których biegunki filtra zdyskretyzowanego wychodzą poza okrąg jednostkowy.

```
In[305]:= PlotPoles[poles_, poles2_] :=
  ListPlot[{Transpose[{Re[poles], Im[poles]}], Transpose[{Re[poles2], Im[poles2]}]}],
  PlotRange -> 1.2 {{-1, 1}, {-1, 1}}, PlotStyle -> {PointSize[Large]},
  Background -> If[Count[poles2, _? (Abs[#] >= 1 &), {1}] > 0, LightRed, White],
  PlotMarkers -> {Automatic, Medium}, AxesOrigin -> {0, 0}, Epilog -> {Green, Circle[]},
  AspectRatio -> Automatic]

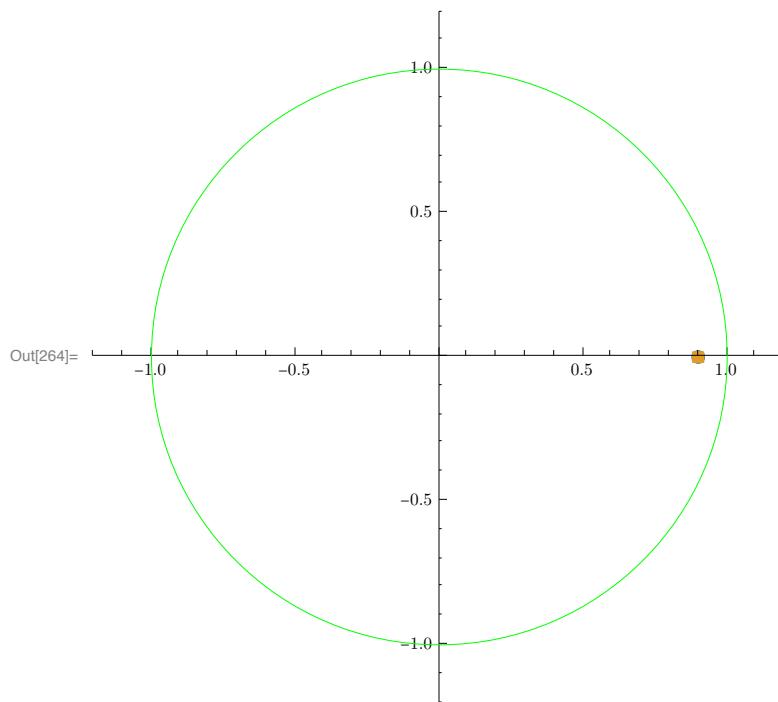
AsymptoticallyStableQ[tfm_?ContinuousTimeModelQ] := If[
  Count[TransferFunctionPoles[tfm], Complex[x_?NonNegative, _] | x_?NonNegative, {3}] > 0,
  False, True]
AsymptoticallyStableQ[tfm_?DiscreteTimeModelQ] := If[
  Count[TransferFunctionPoles[tfm], _? (Abs[#] >= 1 &), {3}] > 0, False, True]
```

★ Testy

```
In[265]:= PlotPoles[{1 - I, I}, {-1 - I, I}]
```



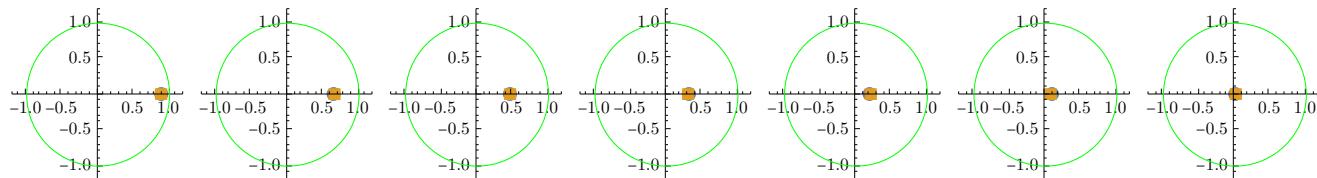
In[264]:= **PlotPoles[polesButt, polesButtD]**

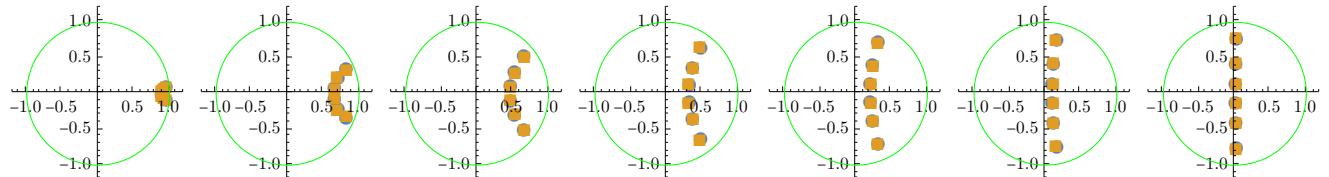
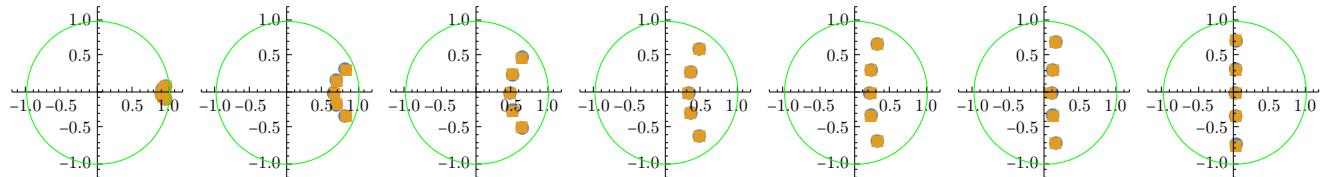
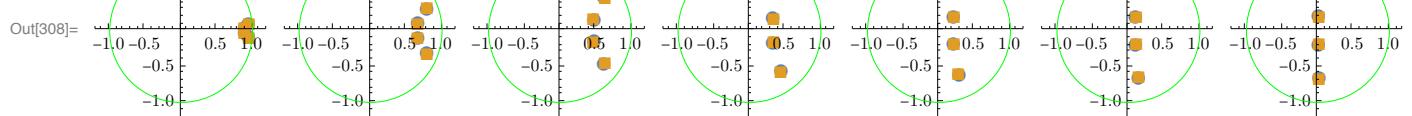
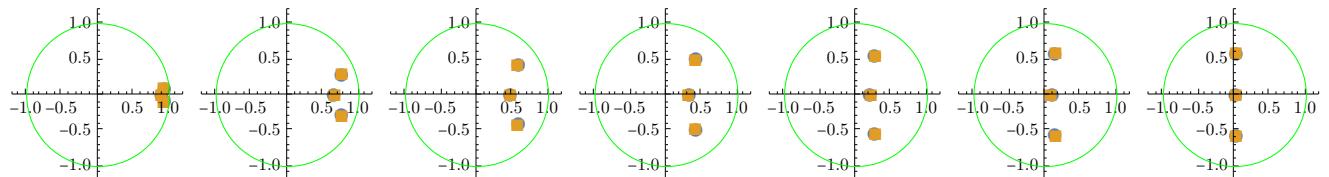
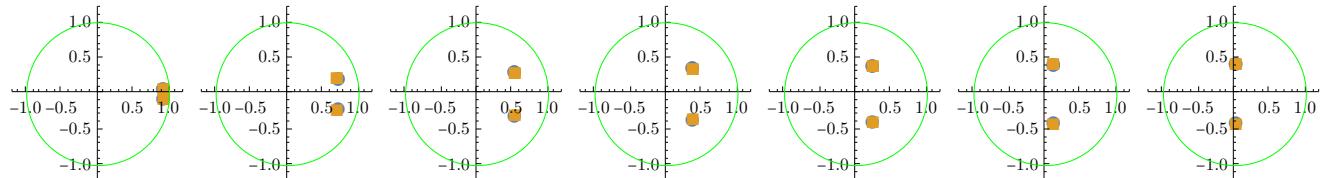


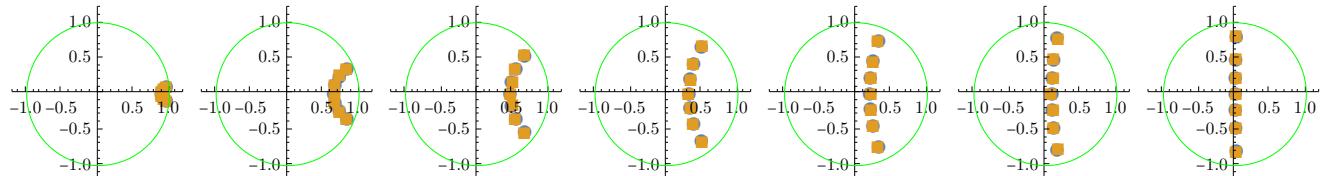
★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

In[308]:= **Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &, {DGbmodels, DGbmodelsD}, 2]**

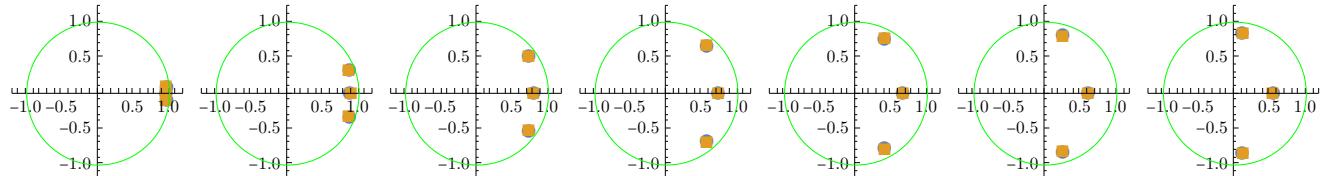
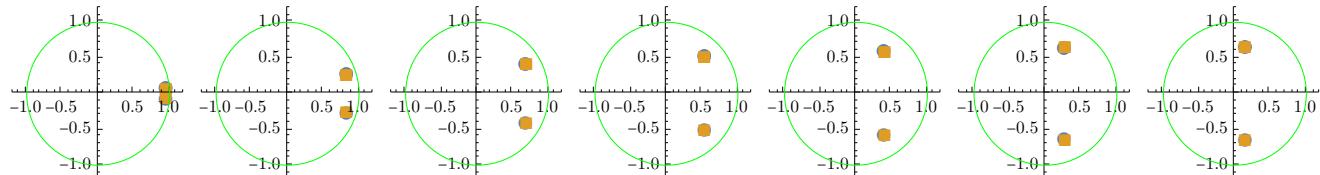
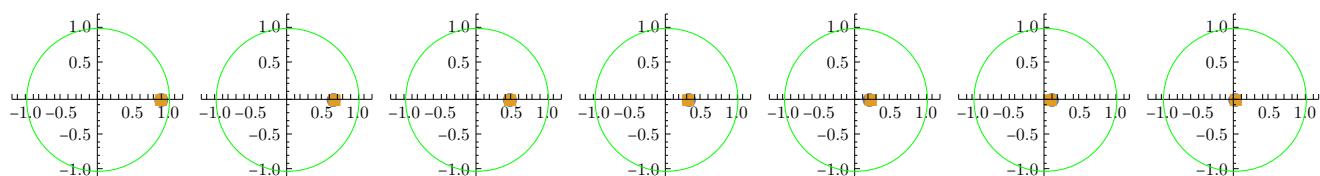


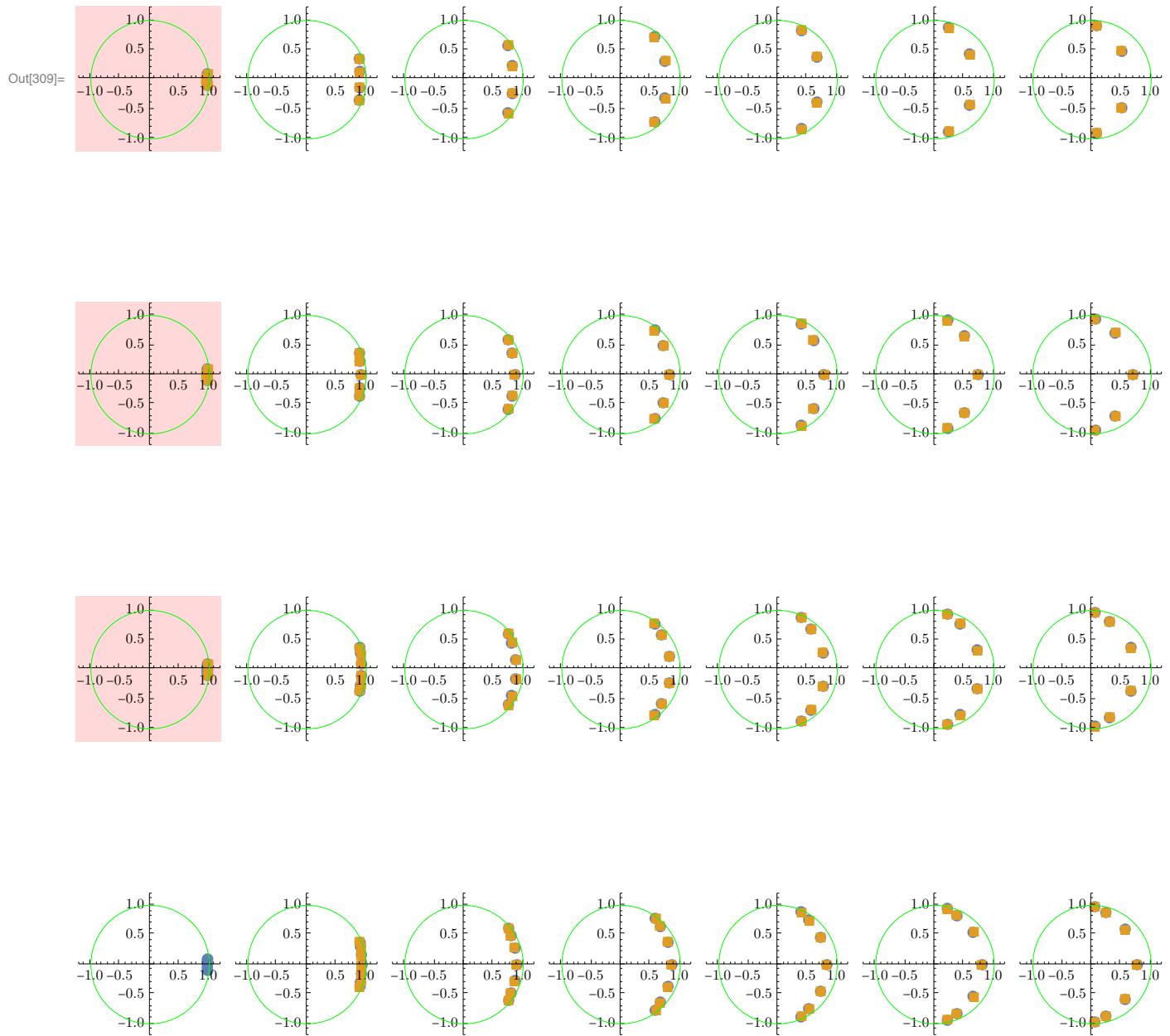




★ Chebyshev 1

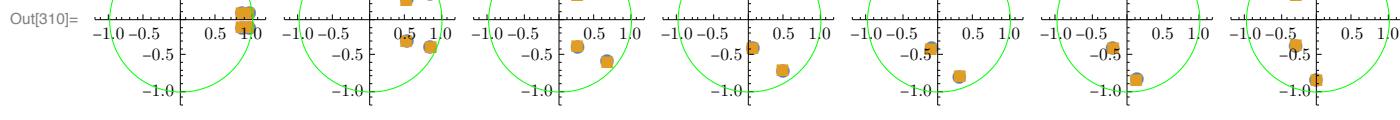
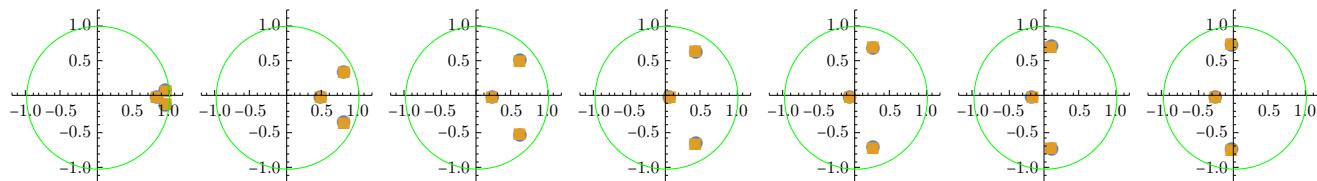
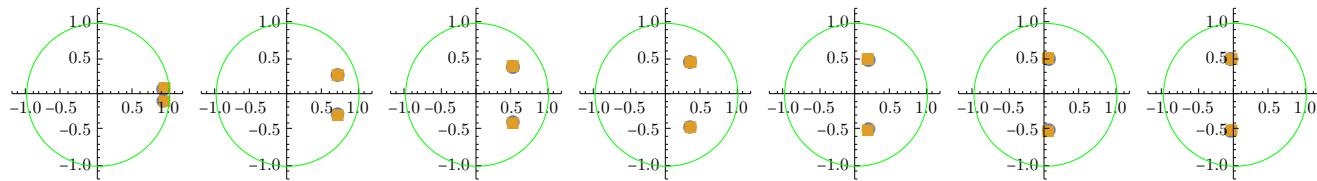
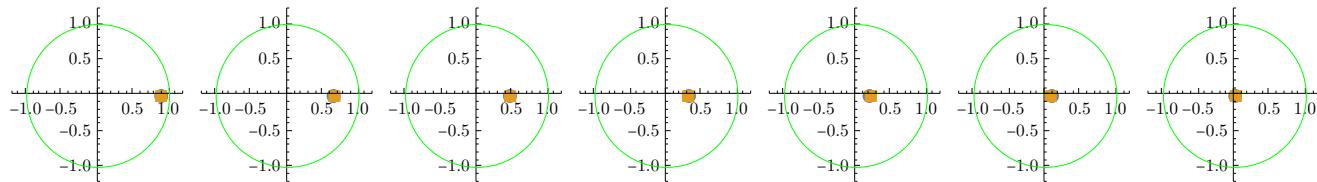
```
In[309]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsD}, 2]
```

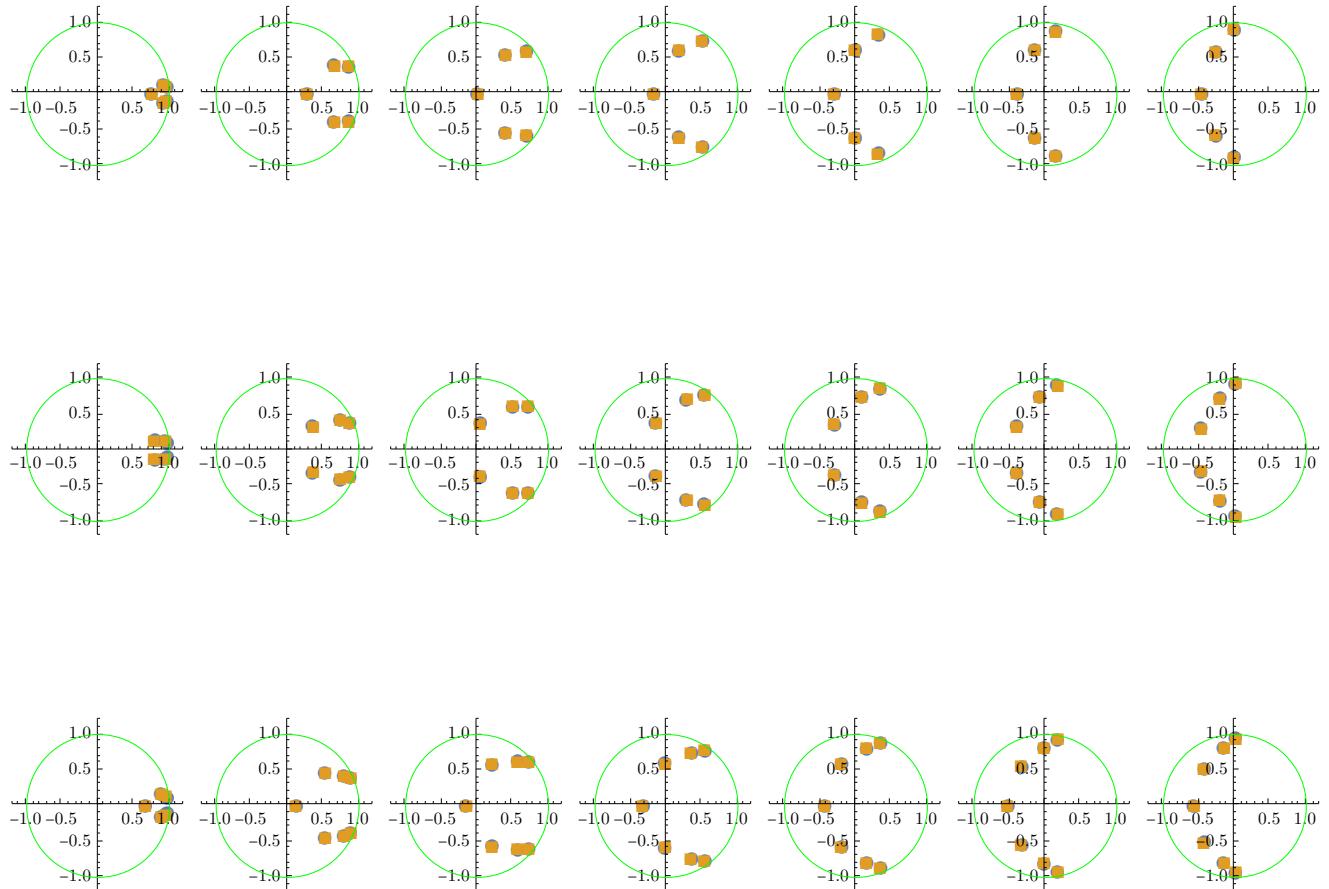




★ Chebyshev 2

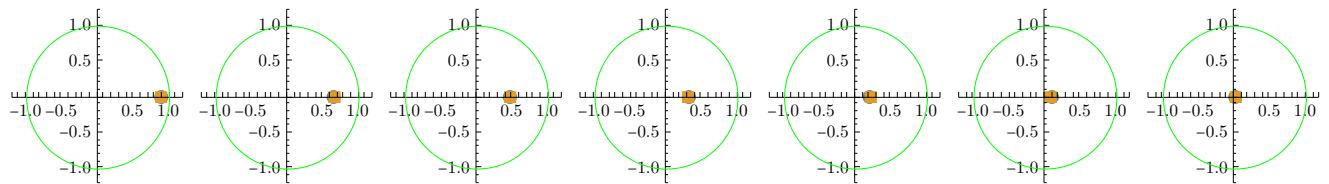
```
In[310]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsD}, 2]
```

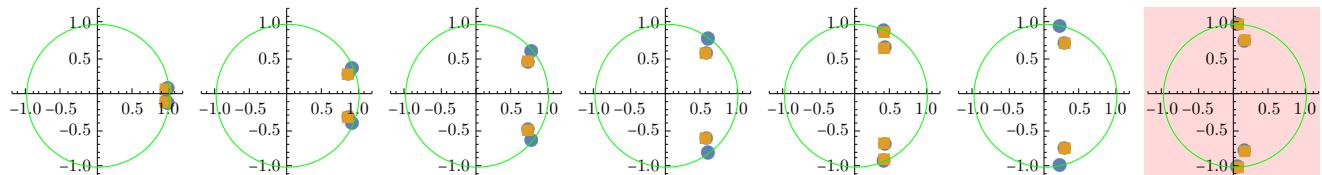
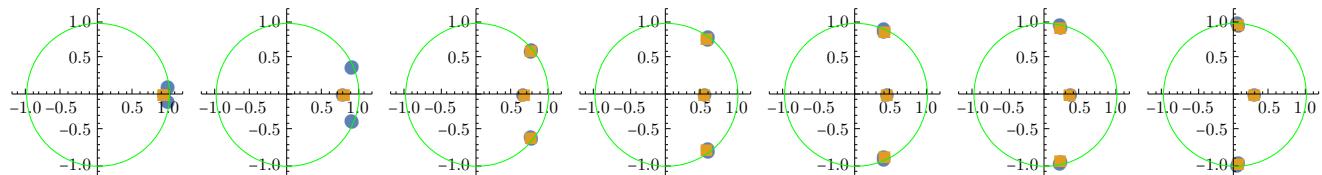
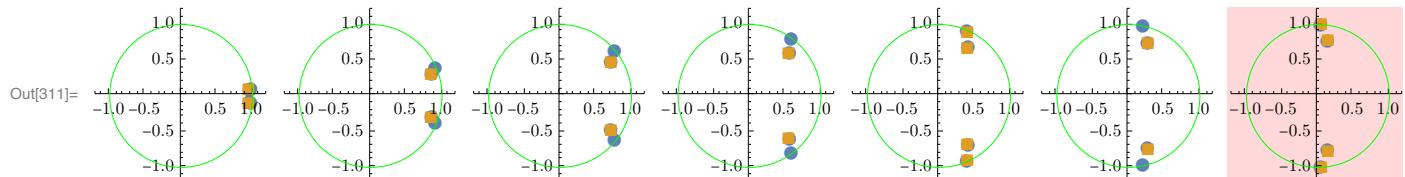
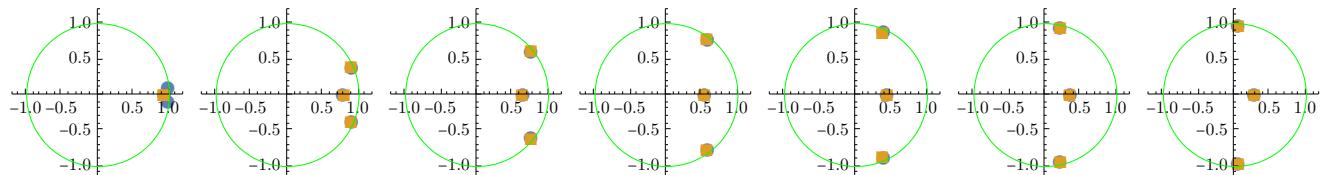
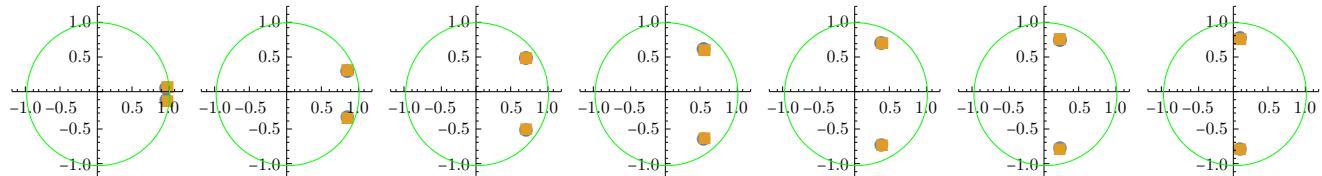


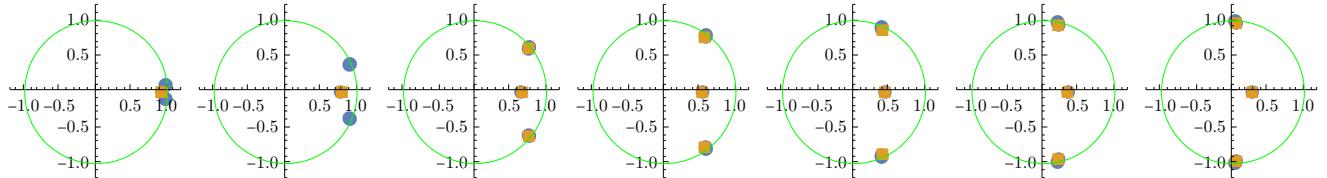


★ Eliptyczne

```
In[311]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsD}, 2]
```







14.3 Dyskretyzacja na poziomie współczynników

14.3.1 Definicje

```
In[359]:= DiscretizeList[x_, max_, bits_] :=
  If[# ≥ 0, 1, -1] Round[(Abs[#]/max) (Power[2, bits - 1] - 1)] max / (Power[2, bits - 1] - 1) & /@ x;
DiscretizeComplexList[x_, y_, bits_] := Module[{xD, yD, coeffD, max},
  If[y == {}, {}, 
    max = Max[Abs[Join[Re@x, Im@x, Re@y, Im@y]]];
    xD = DiscretizeList[Re@x, max, bits] + i DiscretizeList[Im@x, max, bits];
    yD = DiscretizeList[Re@y, max, bits] + i DiscretizeList[Im@y, max, bits];
    {xD, yD}]
  ];
DiscretizeModelCoeffs[tf_, bits_] := Module[{zeros, poles, zerosD, polesD},
  zeros = CoefficientList[Numerator[TransferFunctionExpand[tf][z]][[1, 1]], z];
  poles = CoefficientList[Denominator[TransferFunctionExpand[tf][z]][[1, 1]], z];
  {zerosD, polesD} = DiscretizeComplexList=zeros, poles, bits];
  Chop[FromDigits[Reverse=zerosD], z] / (FromDigits[Reverse[polesD], z])
];
]
```

14.3.2 Testy

```
In[363]:= {a, c} = DiscretizeComplexList[{1, 2, 3}, {i, i2, i3}, 2]
Out[363]= {{0, 3, 3}, {0, 3i, 3i}}
```

Dyskretyzujemy na poziomie filtrów cyfrowych

```
In[371]:= buttModel = TransferFunctionExpand@DGbmmodels[[2, 2]]
```

$$\text{Out}[371]= \left(\frac{0.0302379 + 0.0604758 z + 0.0302379 z^2}{(0.572371 + 0. i) - (1.45142 + 0. i) z + z^2} \right)_1 \mathcal{T}$$

```
In[373]:= polesButt = TransferFunctionPoles[buttModel][[1, 1]]
Out[373]= {0.72571 - 0.213814 i, 0.72571 + 0.213814 i}
```

```
In[374]:= buttModelD = N@DiscretizeModelCoeffs[buttModel, bity]
```

$$\frac{0.04682 + 0.04682 z + 0.04682 z^2}{0.56184 - 1.45142 z + 0.98322 z^2}$$

```
In[375]:= polesButtD = ExtractPoles[buttModelD]
```

$$\text{Out}[375]= \{0.738095 - 0.16323 i, 0.738095 + 0.16323 i\}$$

14.3.3 Dyskretyzacja

```
In[376]:= DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];
```

```
In[377]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];
```

```
In[378]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];
```

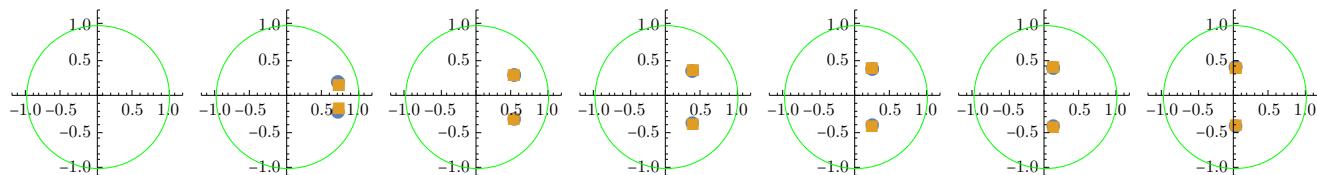
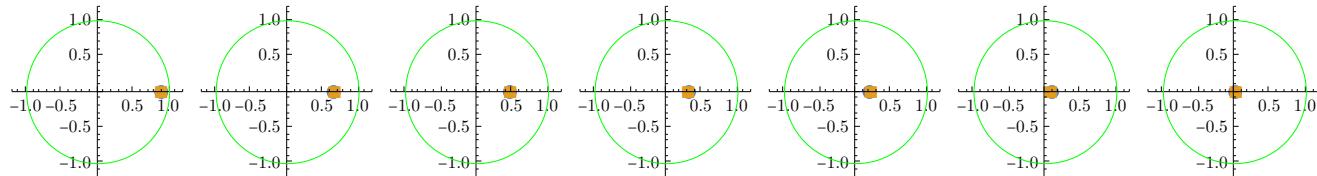
```
In[379]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

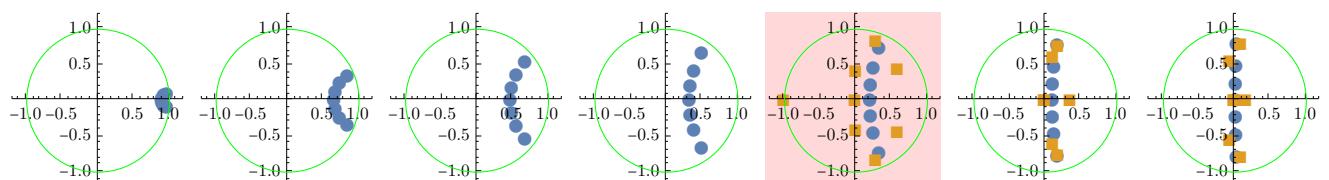
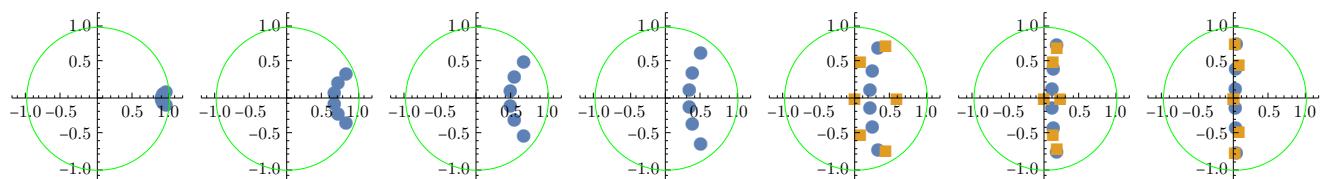
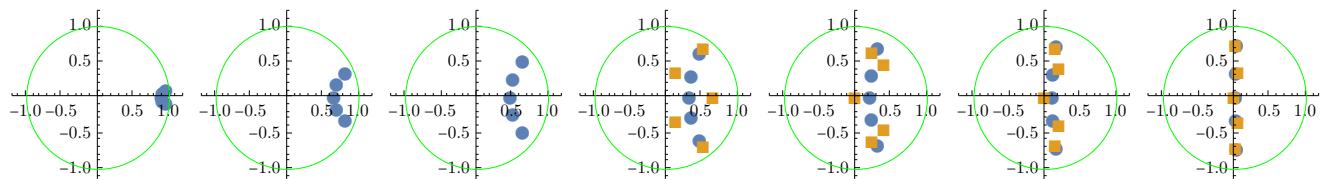
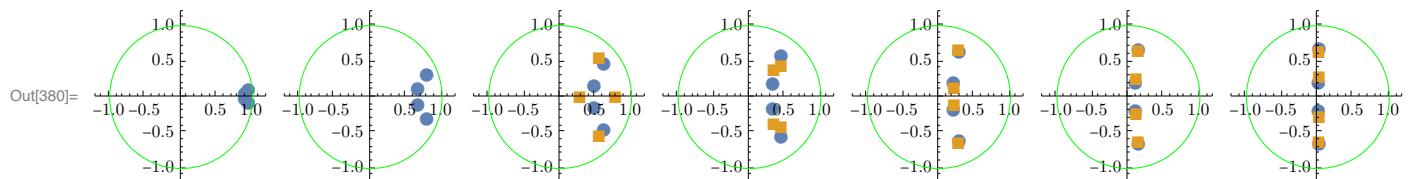
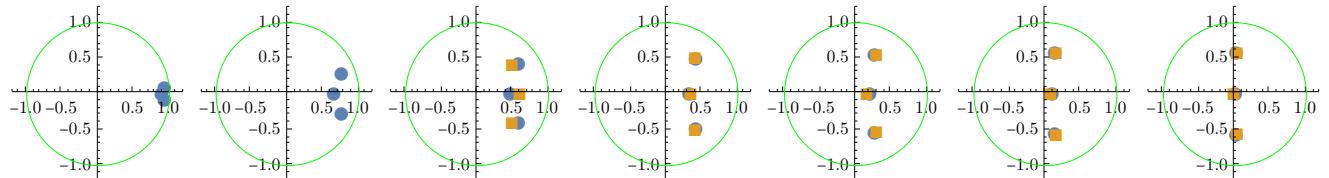
14.3.4 Porównanie położenia biegunków

★ Butterworth

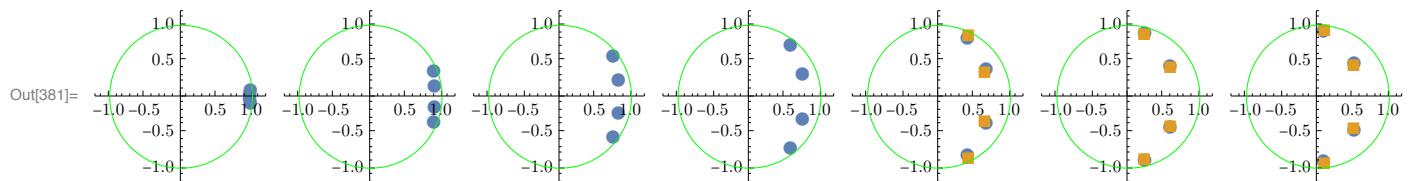
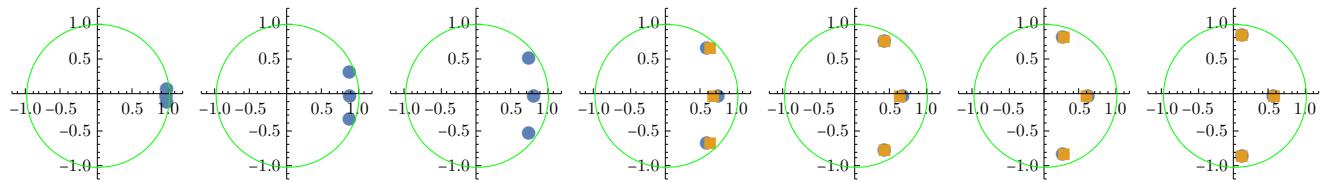
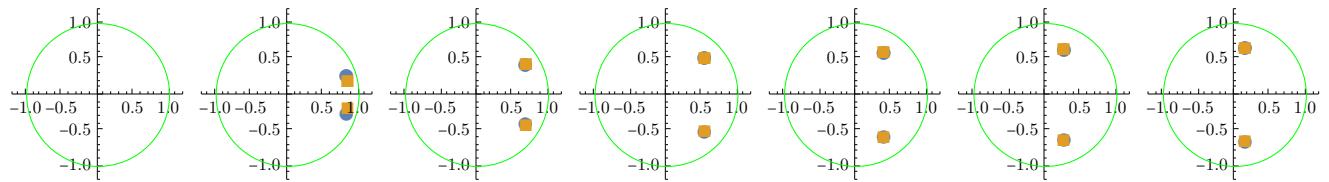
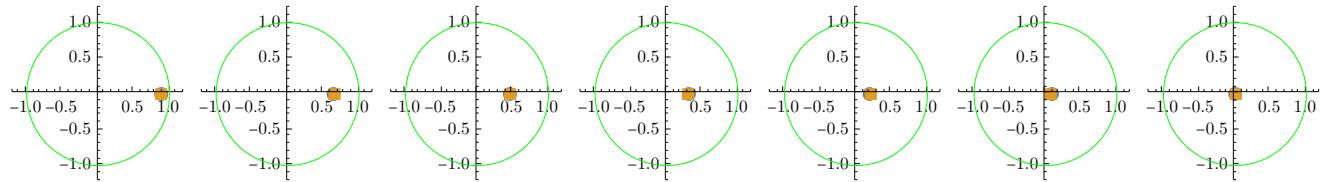
W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

```
In[380]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGmodelsDc}, 2]
```

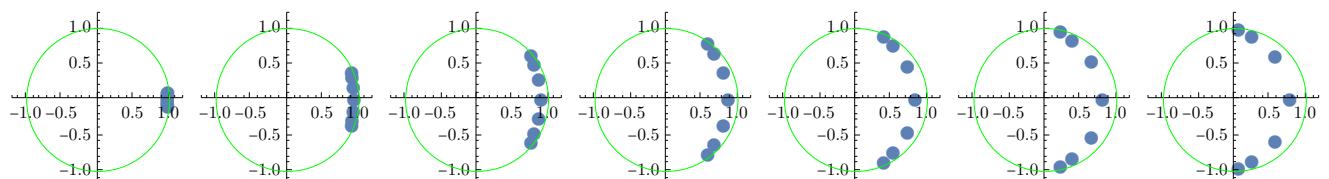
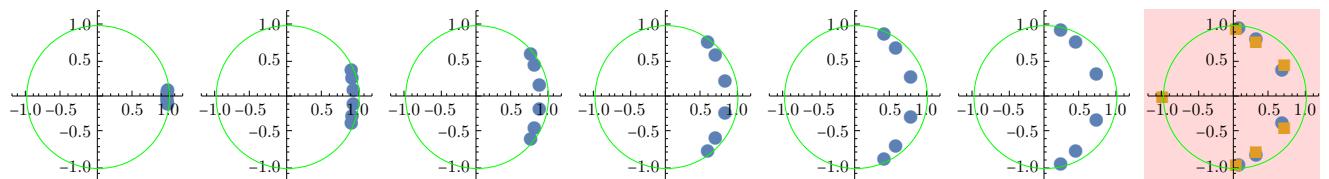
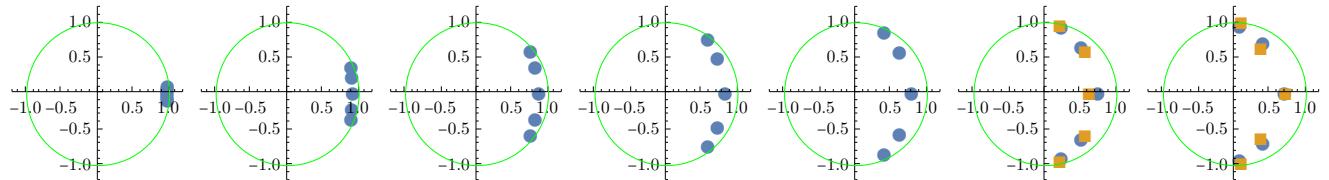




```
In[381]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

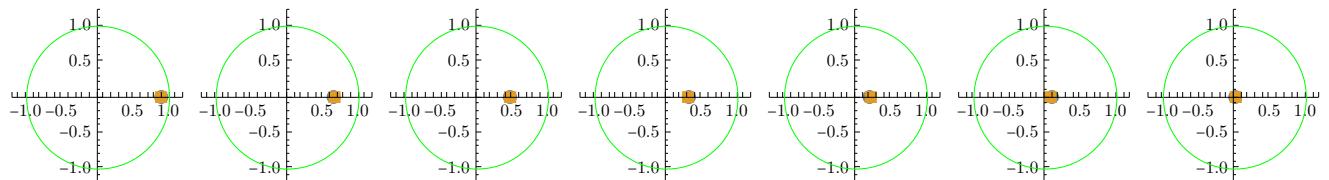


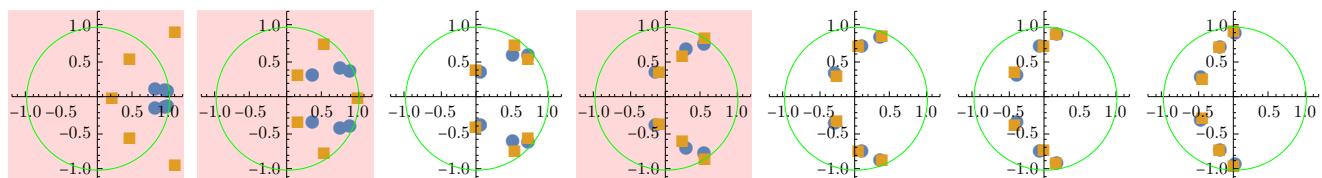
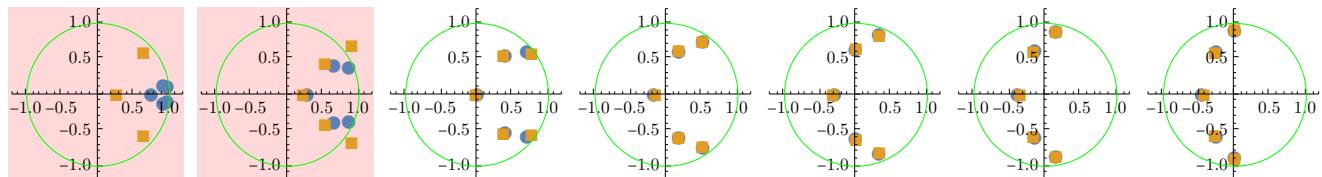
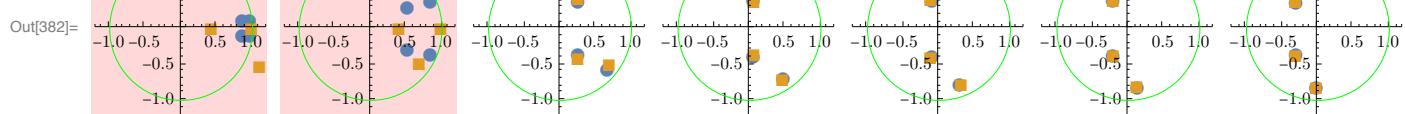
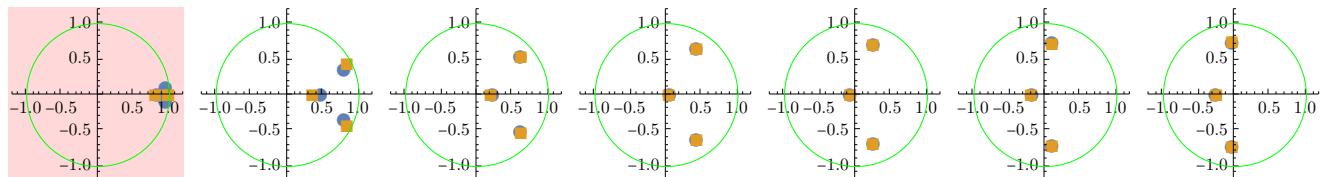
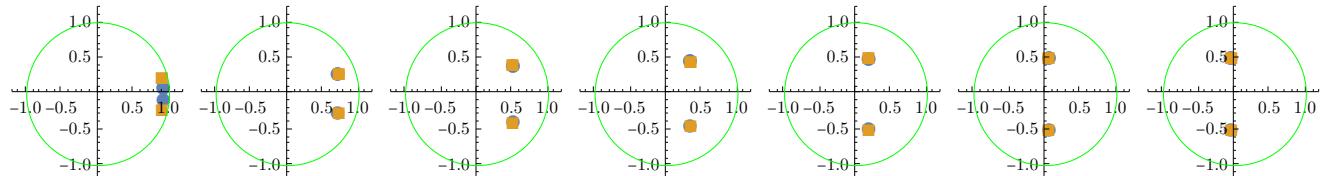
Out[381]=

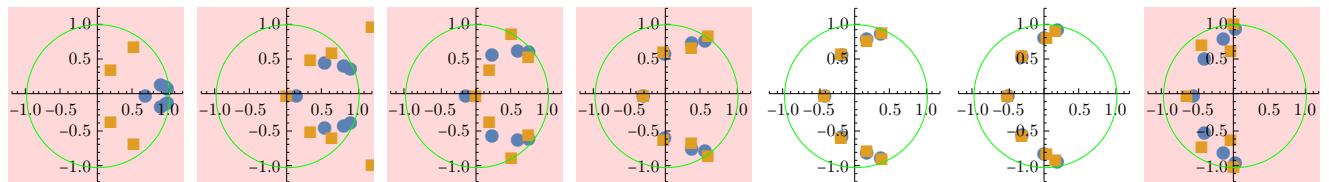


★ Chebyshev 2

```
In[382]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

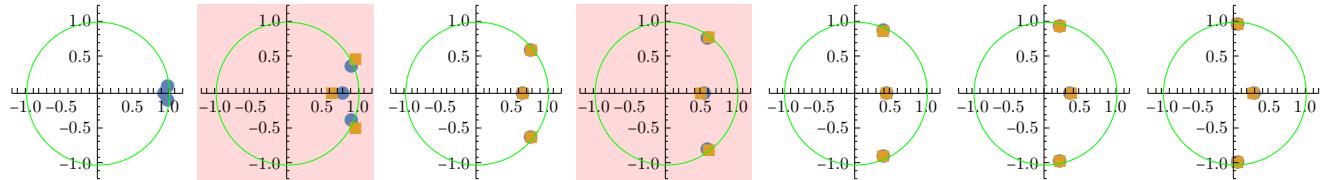
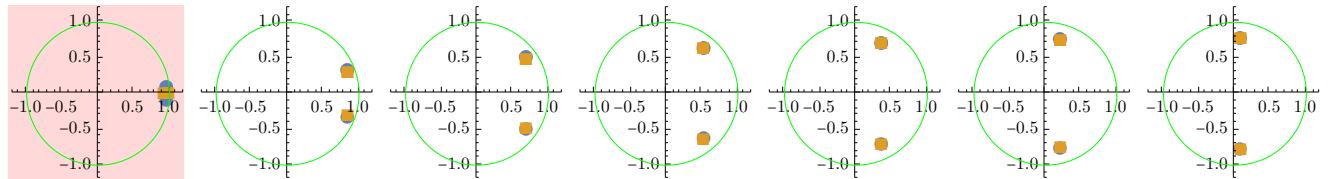
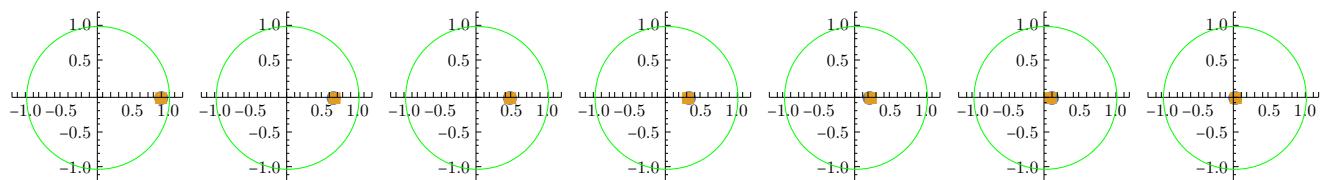


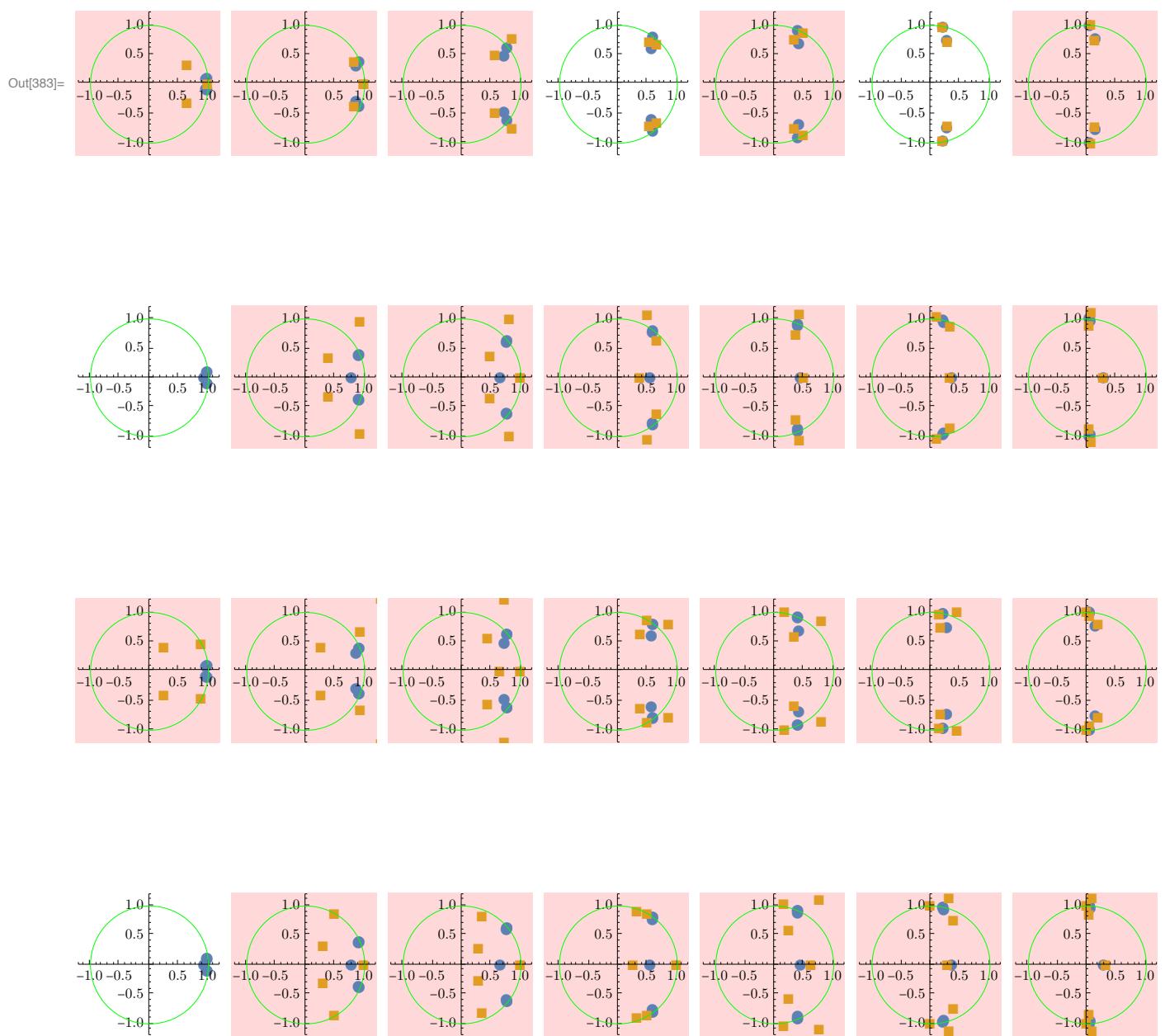




★ Eliptyczne

```
In[383]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





14.4 Dyskusja

Przy dyskretyzacji biegunów koniecznym zabiegiem okazało się nie dyskretyzowanie współczynnika stojącego przed całością filtru, gdyż przy niskich częstotliwościach i wysokich rzędach filtrów stawał się on bardzo mały, a po dyskretyzacji = 0. Przy tych założeniach większość filtrów okazała się stabilna przy dyskretyzacji za pomocą 6 bitów.

Okazuje się jednak, że dyskretyzacja na poziomie biegunów prowadzi do obniżenia rzędu filtru w przypadku filtrów eliptycznych (tj. zera i bieguny znajdujące się blisko siebie zaczynają wzajemnie się

skracać)

Dyskretyzacja współczynników ma bardzo podobny problem widoczny przy niskich częstotliwościach filtru Butterwortha i Chebysheva 1, z powodu znikającego licznika znika cały filtr.

Ponadto widać, iż filtry eliptyczne powyżej 3 rzędu robią się bardzo niestabilne (niektóre nie są zaznaczone na czerwono bo filtr znikł z powodów wspomnianych wyżej)

W świetle tych wyników oraz faktu, iż na urządzeniu będziemy dyskretyzować współczynniki, a nie bieguny, proponuję zastosować dyskretyzację zer i biegunów z osobna, tak by zapobiec zerowaniu się licznika.

Wykonane w celu sprawdzenia tej propozycji symulacje przedstawiam poniżej.

14.5 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

14.5.1 Definicje

Tutaj modyfikuję funkcję DiscretizeComplexList

```
In[422]:= DiscretizeList[x_, max_, bits_] :=
  If[# ≥ 0, 1, -1] Round[(Abs[#] / max) (Power[2, bits - 1] - 1)] max / (Power[2, bits - 1] - 1) & /@ x;
DiscretizeComplexList2[x_, y_, bits_] := Module[{xD, yD, coeffD, max},
  If[y == {}, {}, 
    max = Max[Abs[Join[Re@x, Im@x]]];
    xD = DiscretizeList[Re@x, max, bits] + i DiscretizeList[Im@x, max, bits];
    max = Max[Abs[Join[Re@y, Im@y]]];
    yD = DiscretizeList[Re@y, max, bits] + i DiscretizeList[Im@y, max, bits];
    {xD, yD}]
  ];
  DiscretizeModelCoeffs2[tf_, bits_] := Module[{zeros, poles, zerosD, polesD},
    zeros = CoefficientList[Numerator[TransferFunctionExpand[tf][z]][[1, 1]], z];
    poles = CoefficientList[Denominator[TransferFunctionExpand[tf][z]][[1, 1]], z];
    {zerosD, polesD} = DiscretizeComplexList2=zeros, poles, bits];
    Chop[FromDigits[Reverse=zerosD], z] / (FromDigits[Reverse=polesD], z)]
  ];
```

14.5.2 Testy

```
In[363]:= {a, c} = DiscretizeComplexList[{1, 2, 3}, {i, i 2, i 3}, 2]
Out[363]= {{0, 3, 3}, {0, 3 i, 3 i}}
```

Dyskretyzujemy na poziomie filtrów cyfrowych

```
In[371]:= buttModel = TransferFunctionExpand@DGbmodels[[2, 2]]
```

$$\text{Out}[371]= \left(\frac{0.0302379 + 0.0604758 z + 0.0302379 z^2}{(0.572371 + 0. i) - (1.45142 + 0. i) z + z^2} \right)_1^{\mathcal{T}}$$

```
In[373]:= polesButt = TransferFunctionPoles[buttModel][[1, 1]]
```

$$\text{Out}[373]= \{0.72571 - 0.213814 i, 0.72571 + 0.213814 i\}$$

```
In[374]:= buttModelD = N@DiscretizeModelCoeffs[buttModel, bity]
```

$$\text{Out}[374]= \frac{0.04682 + 0.04682 z + 0.04682 z^2}{0.56184 - 1.45142 z + 0.98322 z^2}$$

```
In[375]:= polesButtD = ExtractPoles[buttModelD]
```

$$\text{Out}[375]= \{0.738095 - 0.16323 i, 0.738095 + 0.16323 i\}$$

14.5.3 Dyskretyzacja

```
In[425]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[426]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[427]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

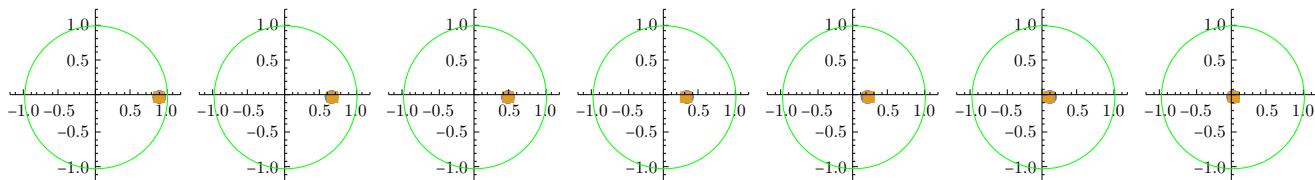
```
In[428]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

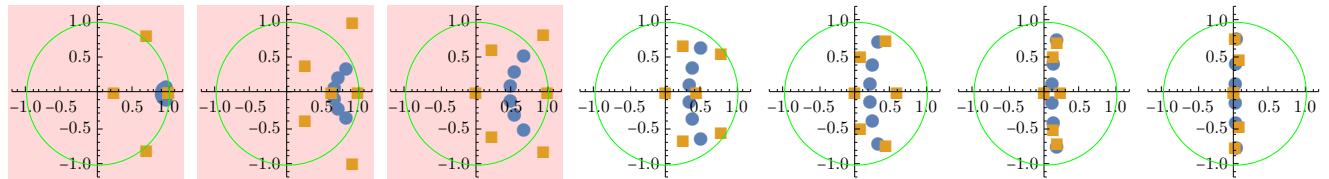
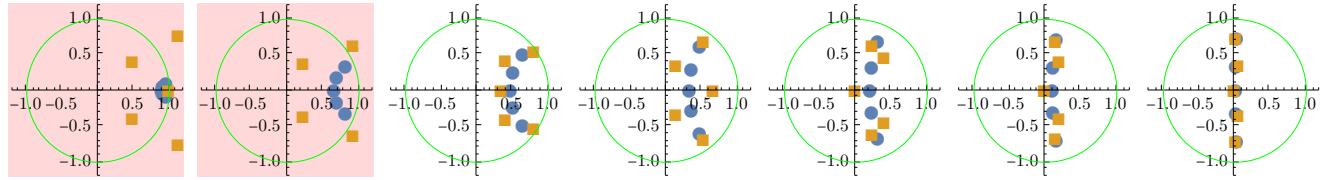
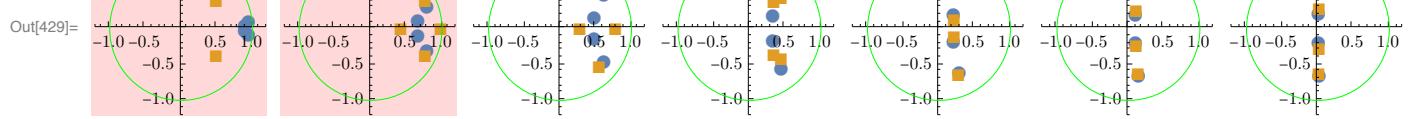
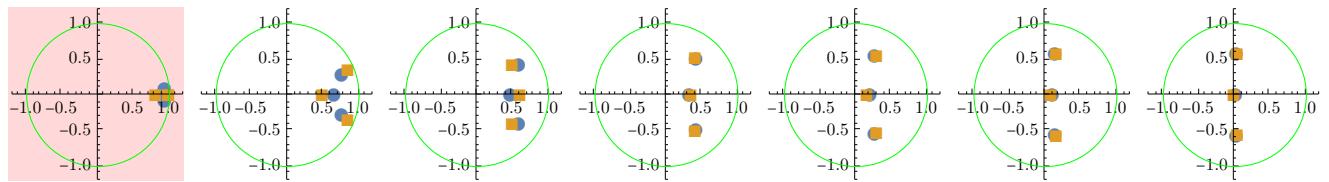
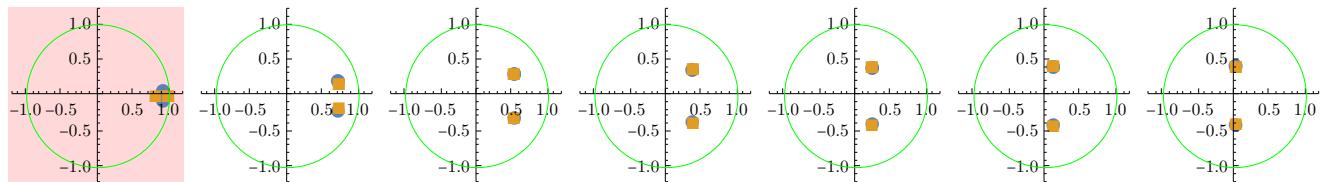
14.5.4 Porównanie położenia biegunków

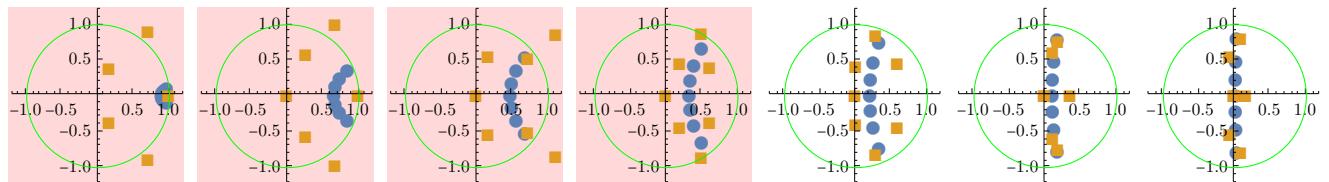
★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

```
In[429]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

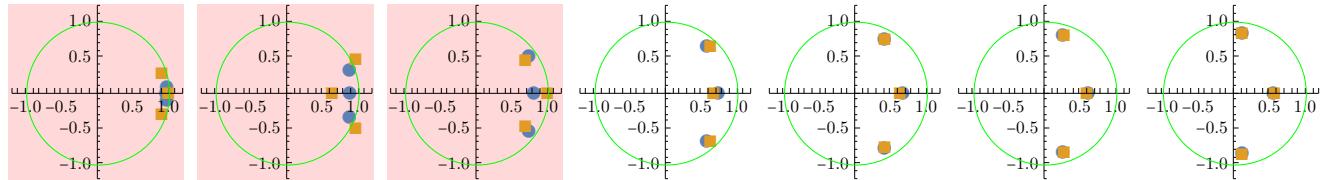
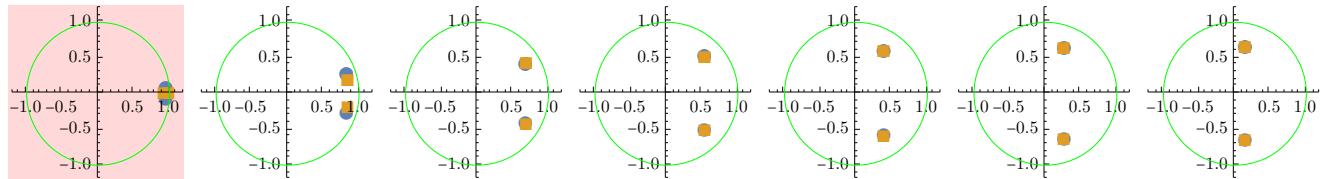
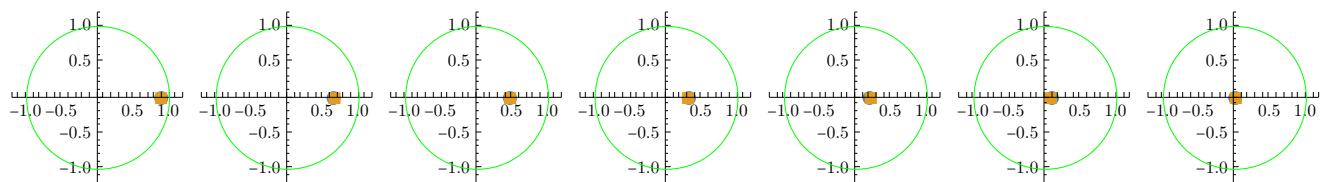


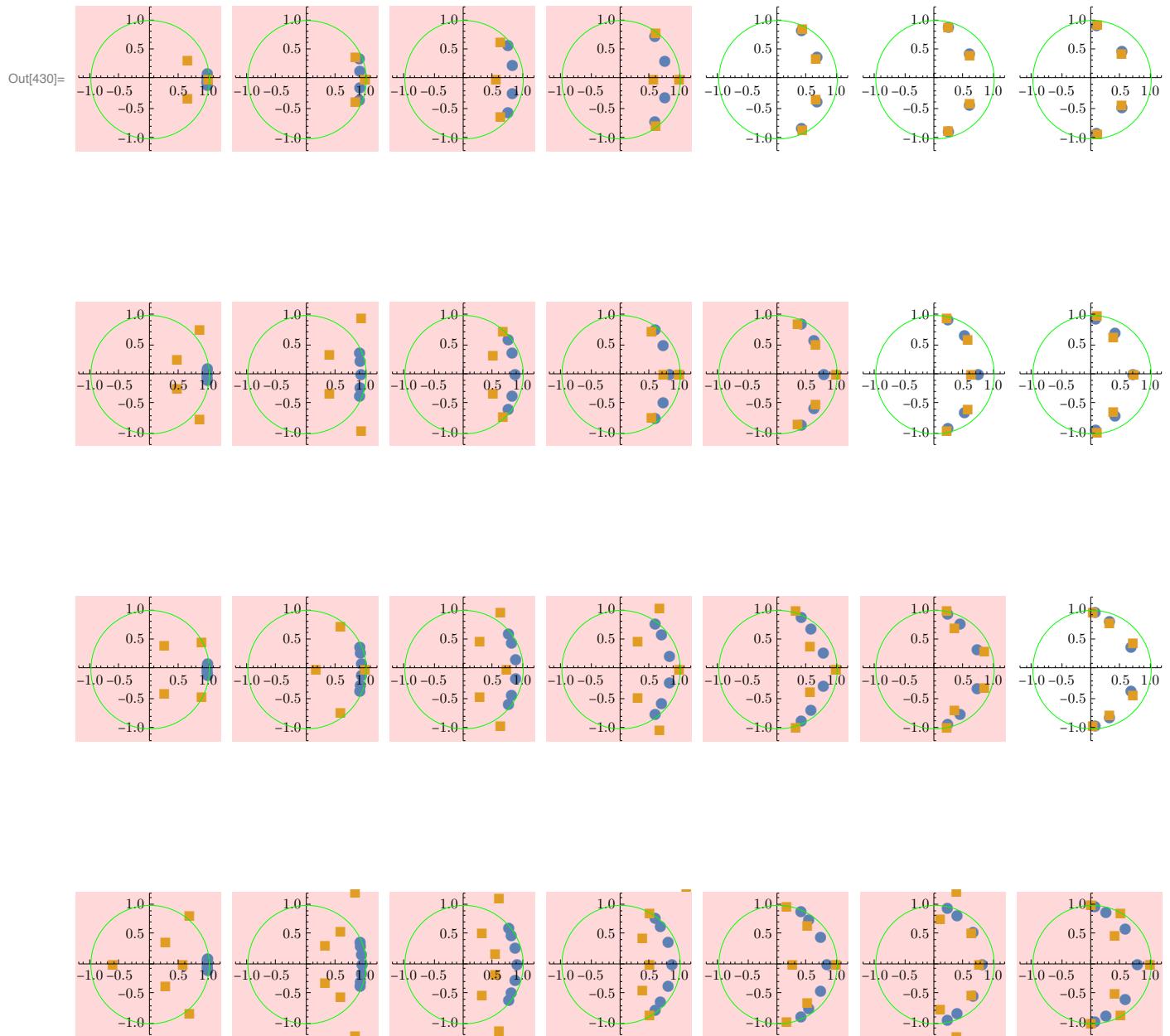




★ Chebyshev 1

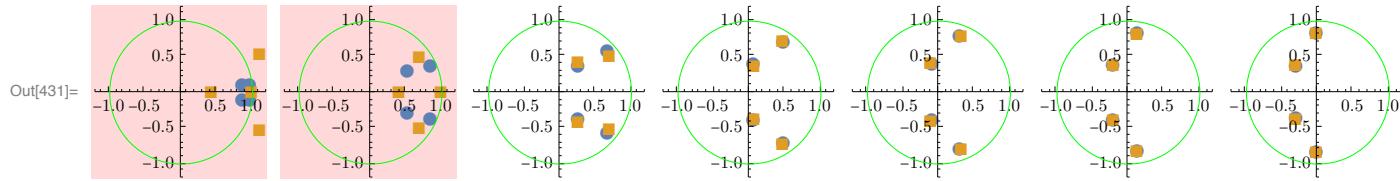
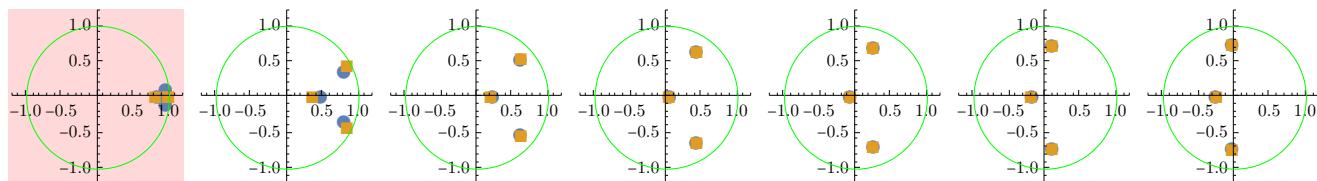
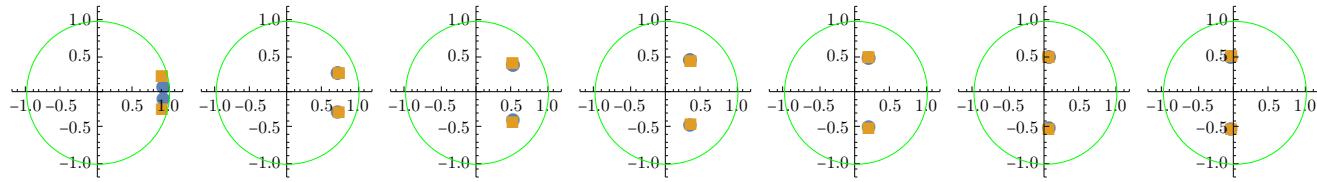
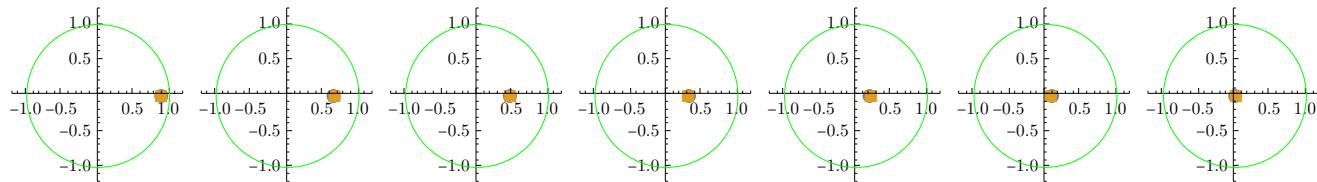
```
In[430]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

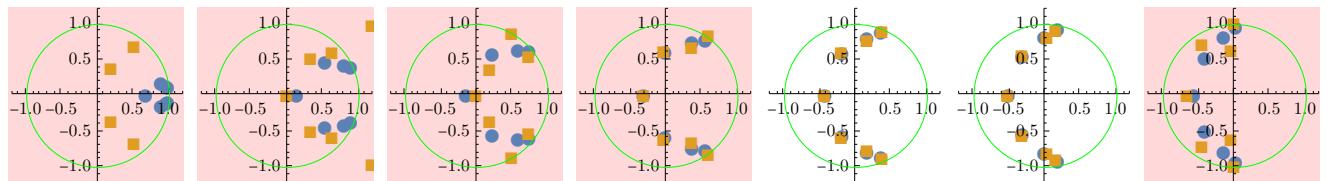
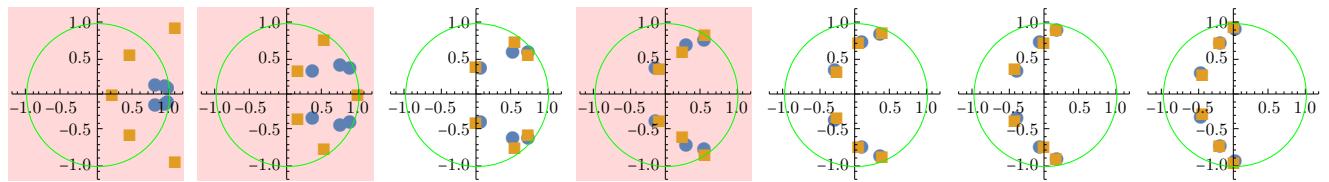
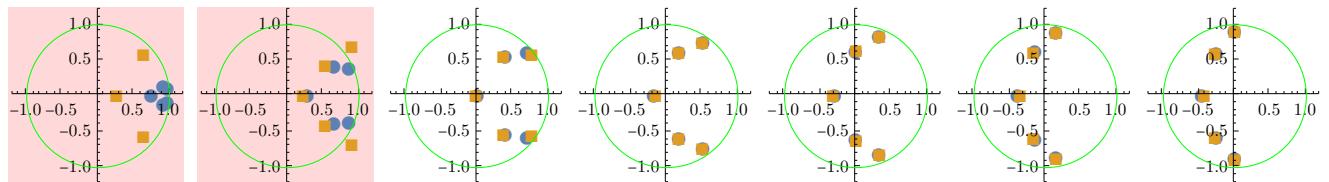




★ Chebyshev 2

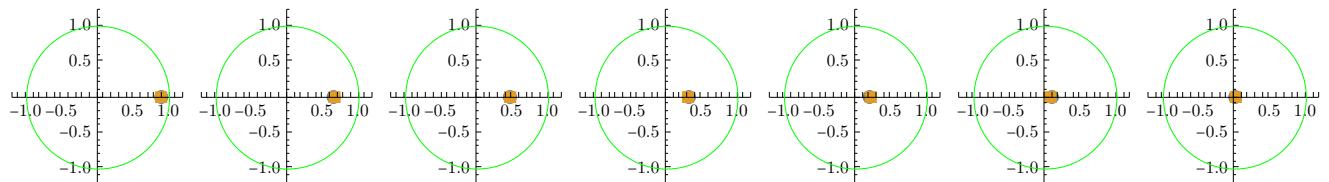
```
In[431]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

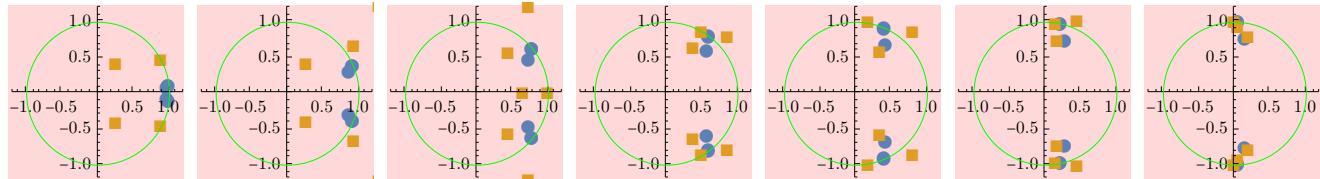
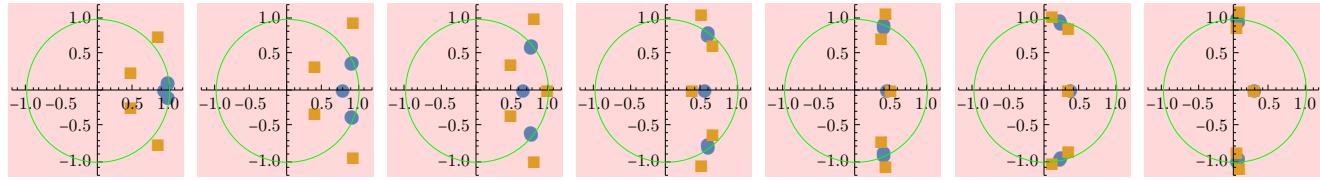
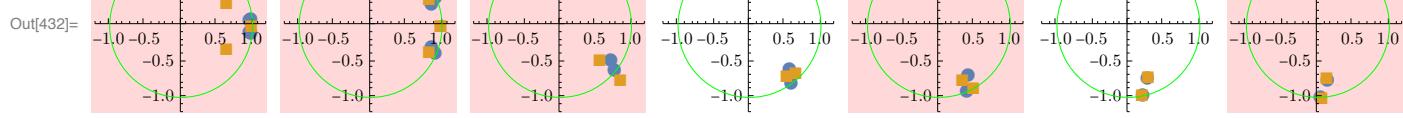
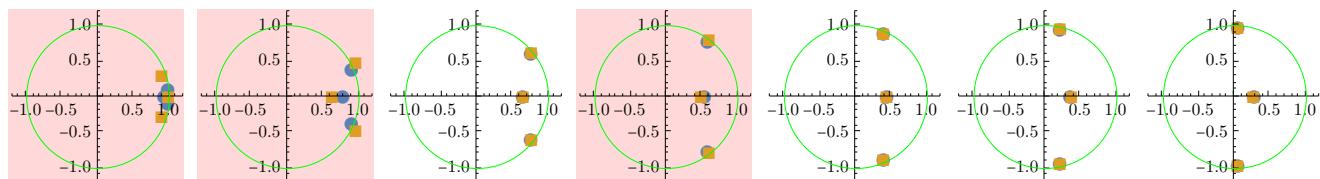
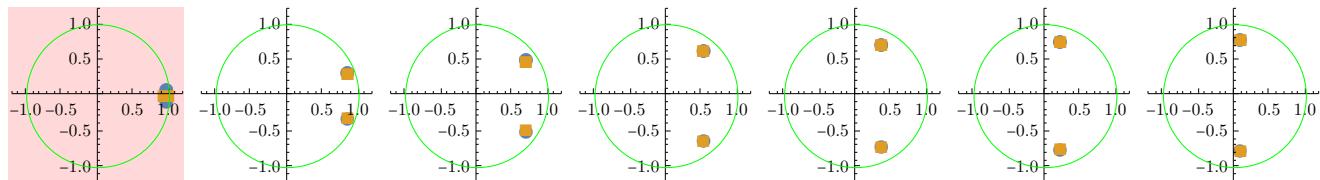


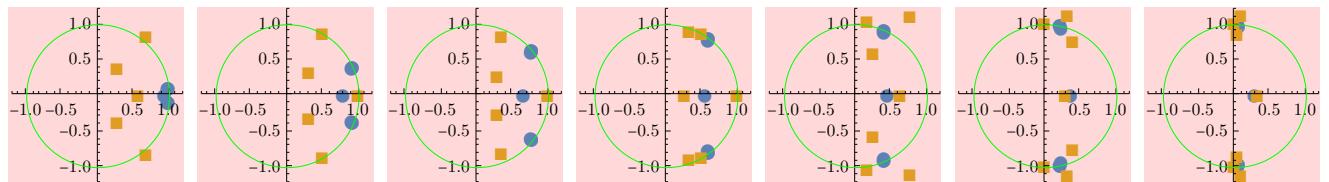


★ Elptyczne

```
In[43]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```







15

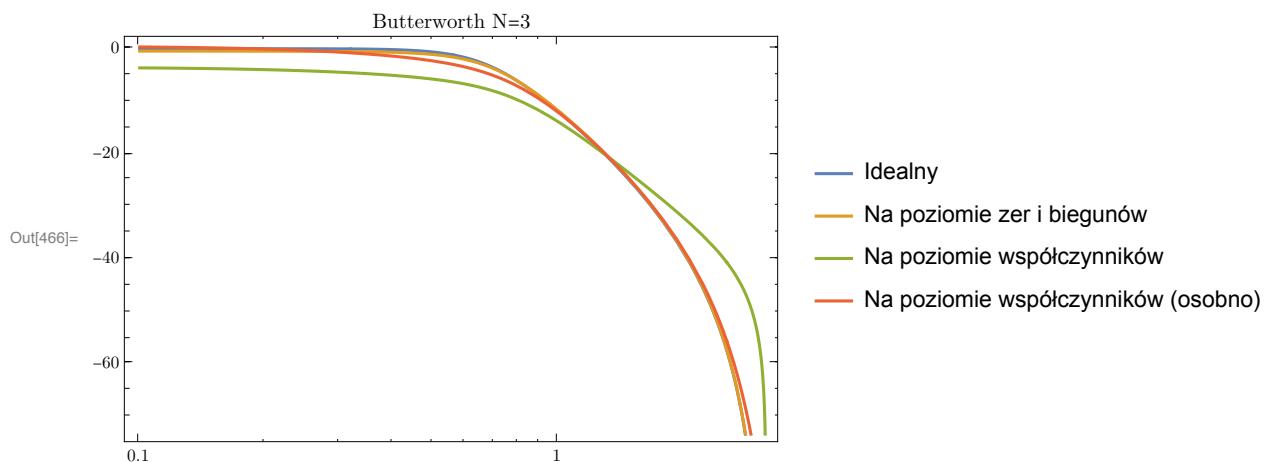
Podsumowanie

Wbrew pozorom podzielenie procesu dyskretyzacji na współczynniki licznika i mianownika nie przyniosło wielkich zmian - filtry który znikły, stały się po prostu niestabilne.

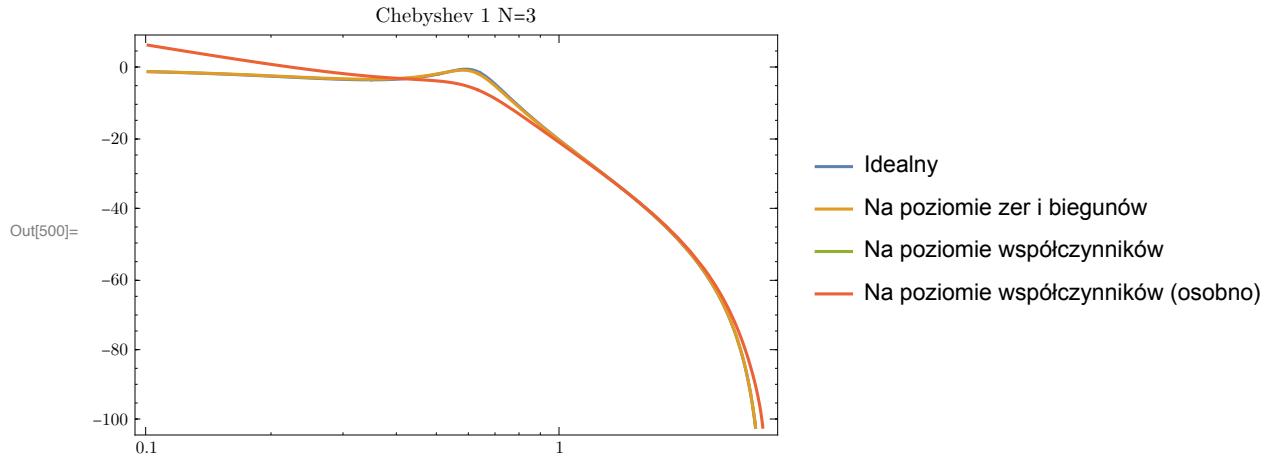
Używając Bode Plot (ilustrujący kwadrat amplitudy (dB) w zależności od ω) można zademonstrować, że lepsze odwzorowanie filtru zależy właśnie od przyjętej strategii dyskretyzacji współczynników.

Poniżej znajdują się przykładowe Bode Ploty dla filtrów Butterwortha trzeciego rzędu i częstotliwości równej 1.

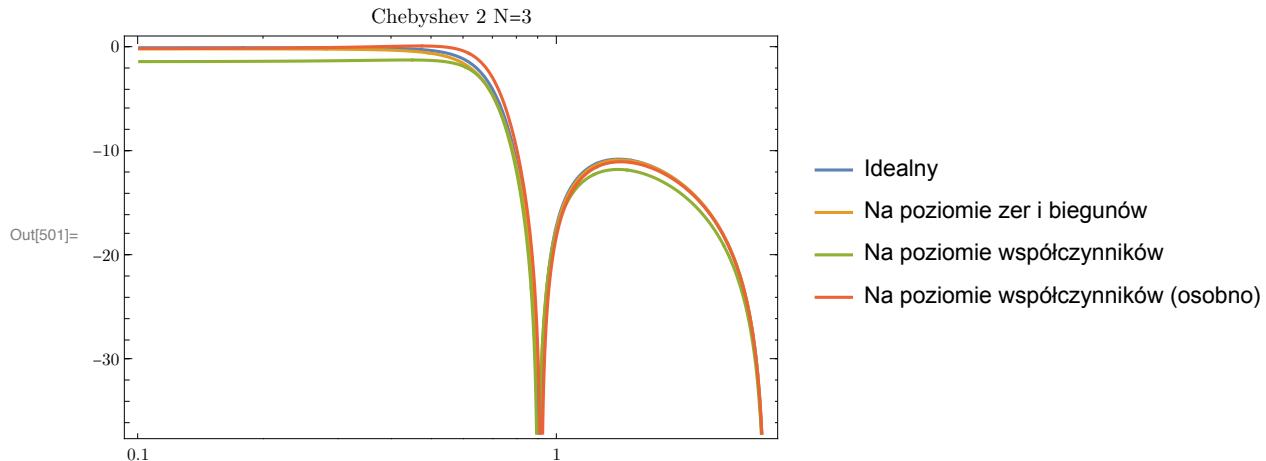
```
In[466]:= BodePlot[{DGbmodels[[3, 3]][eiω], DGbmodelsD[[3, 3]] /. {z -> eiω},
  DGbmodelsDc[[3, 3]] /. {z -> eiω}, DGbmodelsDc2[[3, 3]] /. {z -> eiω}], {ω, 0.1, π},
  PlotLayout -> "Magnitude",
  PlotLegends -> {"Idealny", "Na poziomie zer i biegunów", "Na poziomie współczynników",
  "Na poziomie współczynników (osobno)"}, PlotLabel -> "Butterworth N=3"]
```



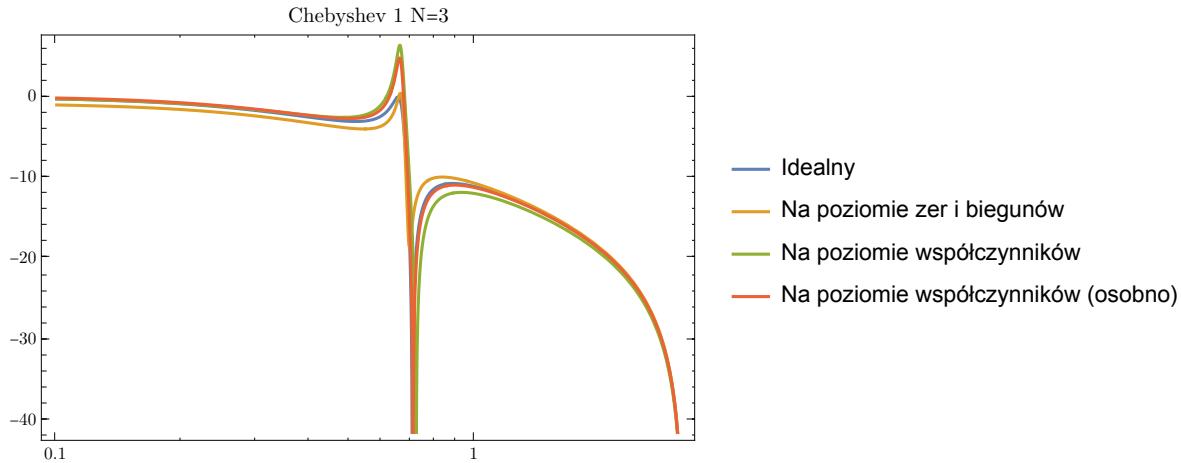
```
In[500]:= BodePlot[{DGc1models[[3, 3]][ei ω], DGc1modelsD[[3, 3]] /. {z -> ei ω},  
DGc1modelsDc[[3, 3]] /. {z -> ei ω}, DGc1modelsDc2[[3, 3]] /. {z -> ei ω}}, {ω, 0.1, π},  
PlotLayout -> "Magnitude",  
PlotLegends -> {"Idealny", "Na poziomie zer i biegunów", "Na poziomie współczynników",  
"Na poziomie współczynników (osobno)"}, PlotLabel -> "Chebyshev 1 N=3"]
```



```
In[501]:= BodePlot[{DGc2models[[3, 3]][ei ω], DGc2modelsD[[3, 3]] /. {z -> ei ω},  
DGc2modelsDc[[3, 3]] /. {z -> ei ω}, DGc2modelsDc2[[3, 3]] /. {z -> ei ω}}, {ω, 0.1, π},  
PlotLayout -> "Magnitude",  
PlotLegends -> {"Idealny", "Na poziomie zer i biegunów", "Na poziomie współczynników",  
"Na poziomie współczynników (osobno)"}, PlotLabel -> "Chebyshev 2 N=3"]
```



```
In[502]:= BodePlot[{DGmodels[[3, 3]][ei ω], DGmodelsD[[3, 3]] /. {z -> ei ω},  
DGmodelsDc[[3, 3]] /. {z -> ei ω}, DGmodelsDc2[[3, 3]] /. {z -> ei ω}], {ω, 0.1, π},  
PlotLayout -> "Magnitude",  
PlotLegends -> {"Idealny", "Na poziomie zer i biegunów", "Na poziomie współczynników",  
"Na poziomie współczynników (osobno)"}, PlotLabel -> "Chebyshev 1 N=3"]
```



Jeśli trzeba można wyrysować wszystkie tym poleceniem.

```
Grid@  
MapThread[BodePlot[{#1[ei ω], #2 /. {z -> ei ω}, #3 /. {z -> ei ω}, #4 /. {z -> ei ω}],  
{ω, 0.01, π}, PlotLayout -> "Magnitude"] &, {DGbmodels, DGbmodelsD, DGbmodelsDc, DGbmodelsDc2},  
2]
```

Out[440]= \$Aborted

Wybrana przeze mnie wcześniej procedura dyskretyzacji na poziomie zer i biegunów (dodatkowo wcześniej robiłem to na poziomie filtru analogowego), pomimo swoich pozornych zalet obarczona jest jednak dodatkowym błędem, wynikającym z wymnożenia zdyskretyzowanych zer/biegunów którego należało by dokonać nie na liczbach rzeczywistych, lecz na liczbach n-bitowych.

Prawdopodobnie to dyskwalifikuje tę metodę.

Zatem mając dwóch kandydatów (na poziomie współczynników, i na poziomie współczynników - osobno zera i bieguny) do metody dyskretyzacji przeprowadzam symulacje dla wyższej liczby bitów.

16 Badanie stabilności filtrów cyfrowych (7 bit)

In[524]:= **bity** = 7;

16.1 Dyskretyzacja na poziomie współczynników

16.1.1 Dyskretyzacja

In[525]:= **DGbmodelsDc** = Map[DiscretizeModelCoeffs[#, bity] &, **DGbmodels**, {2}];

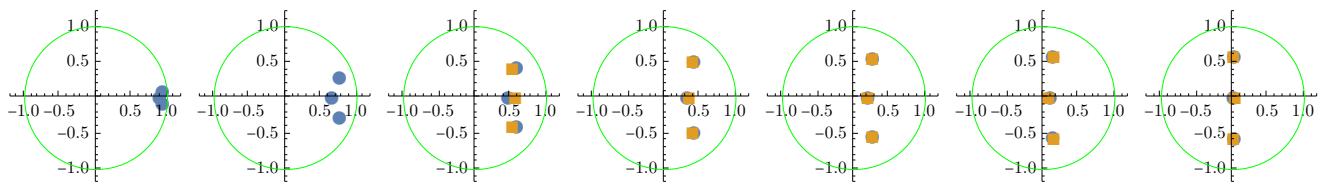
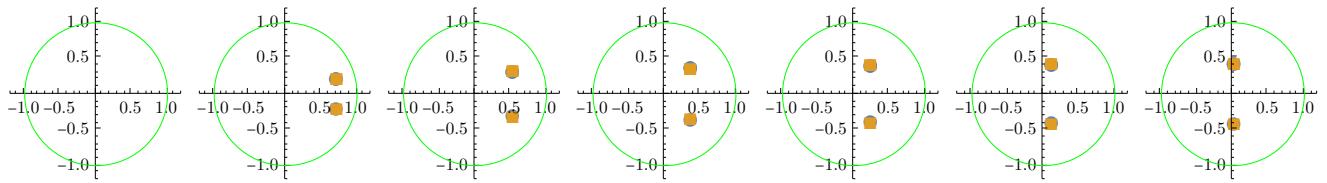
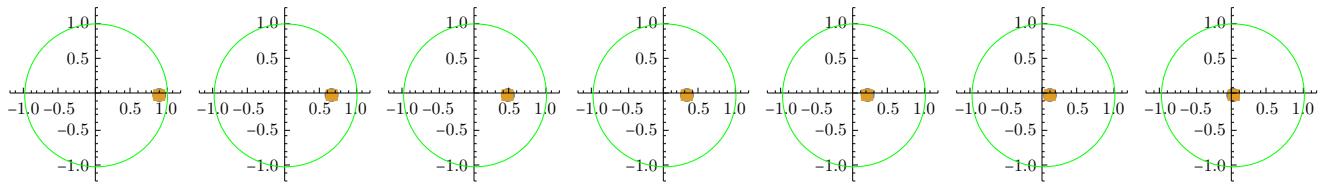
```
In[526]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[527]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[528]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

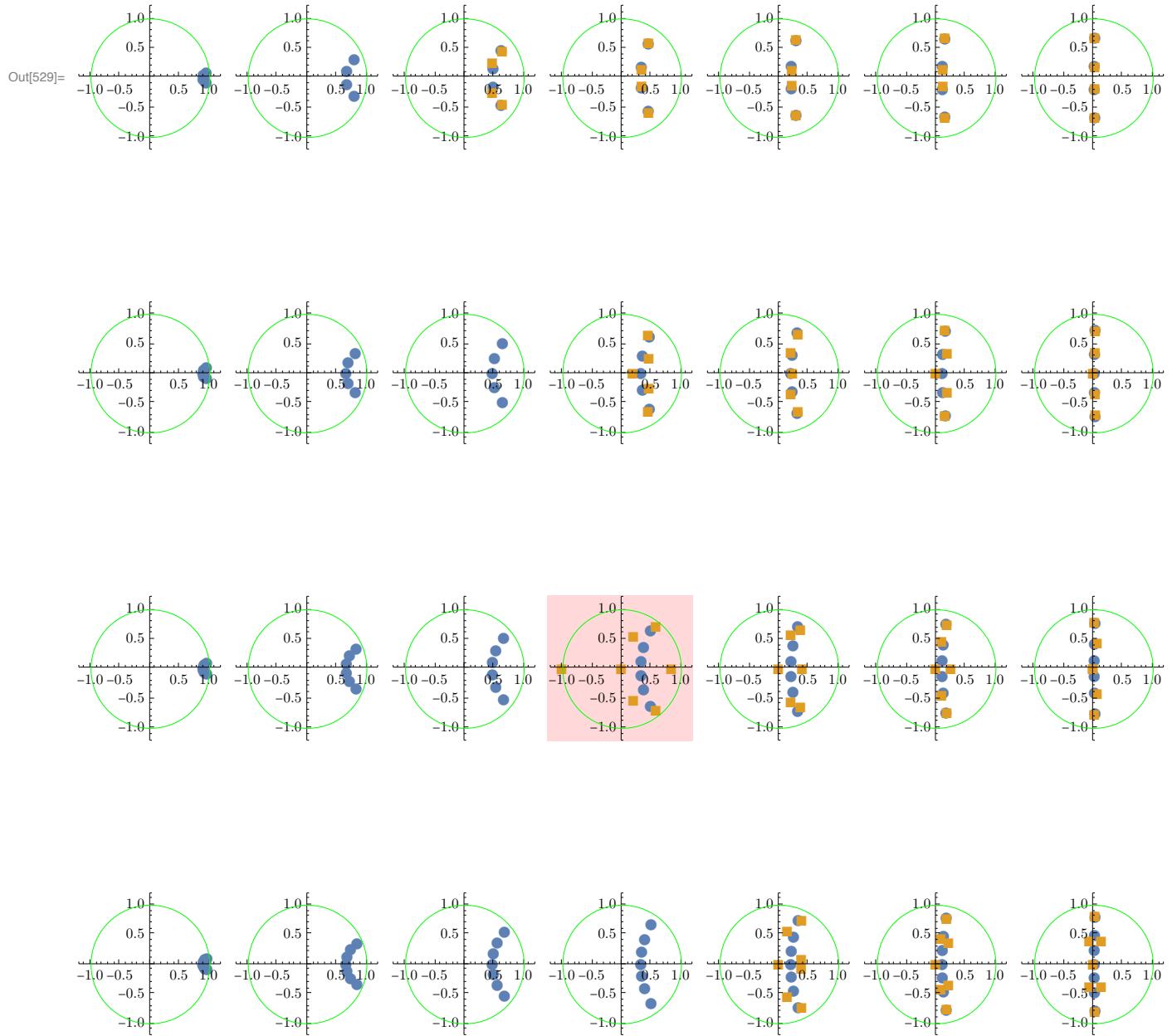
16.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

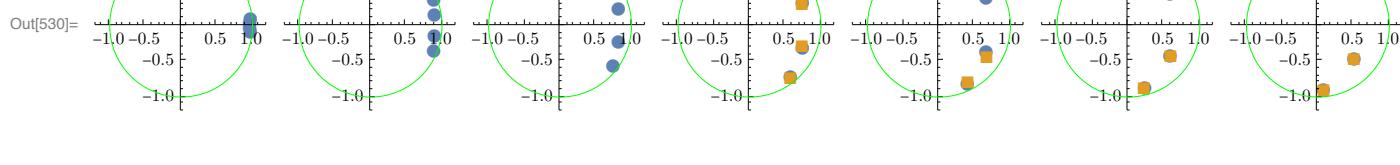
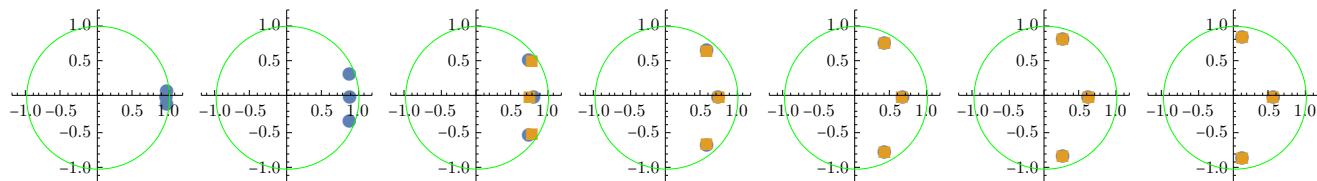
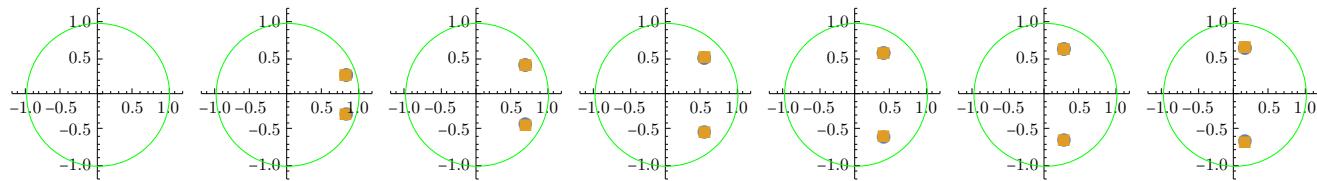
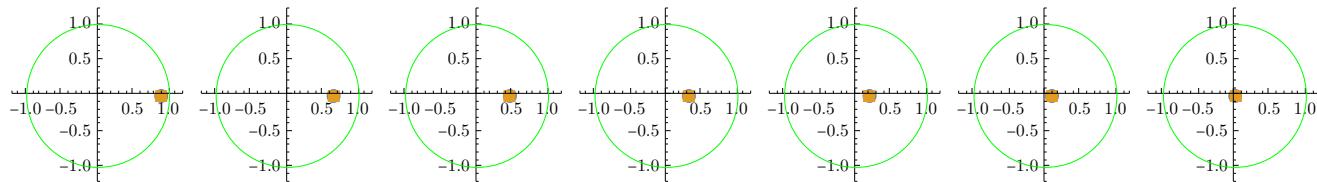
```
In[529]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

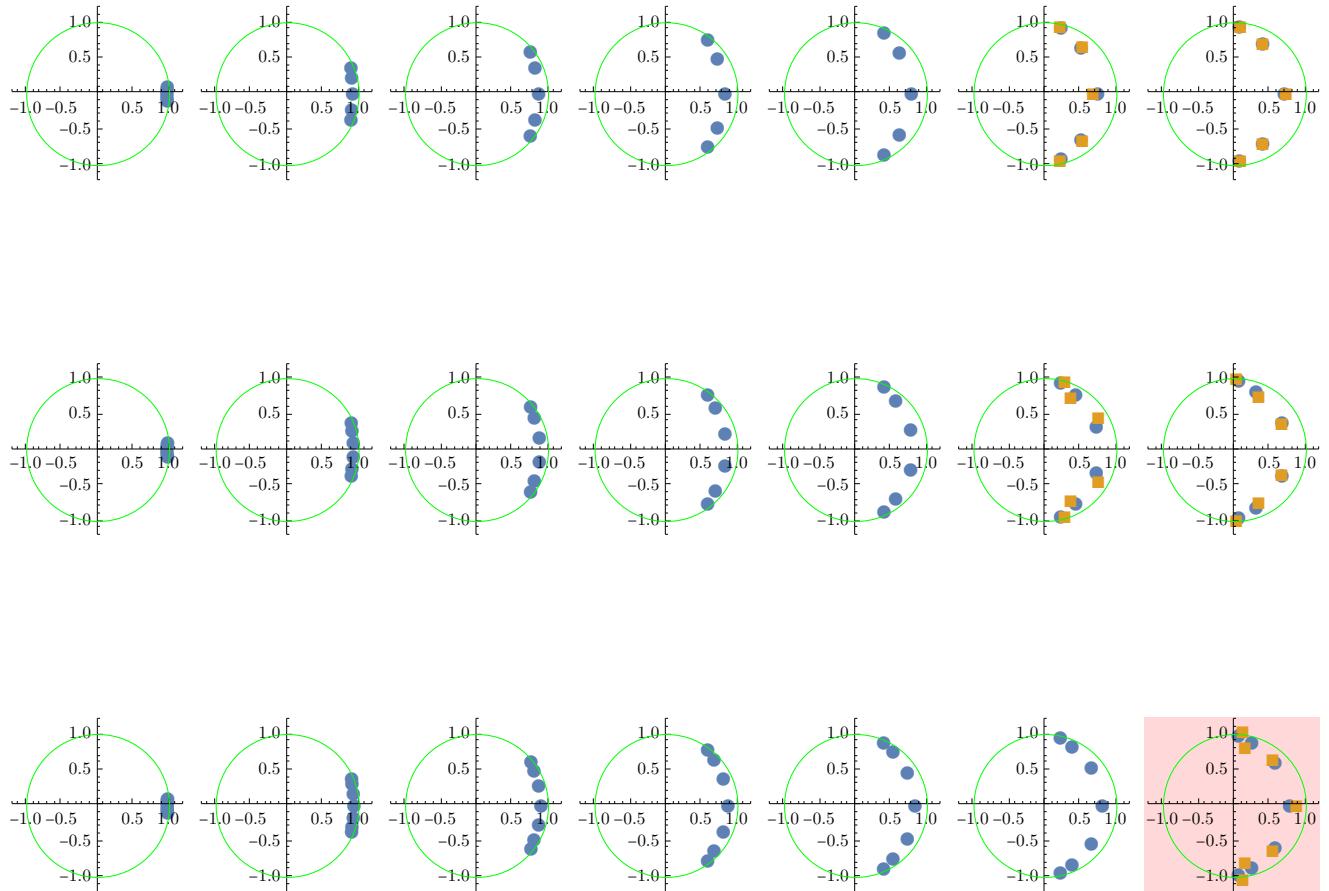




★ Chebyshev 1

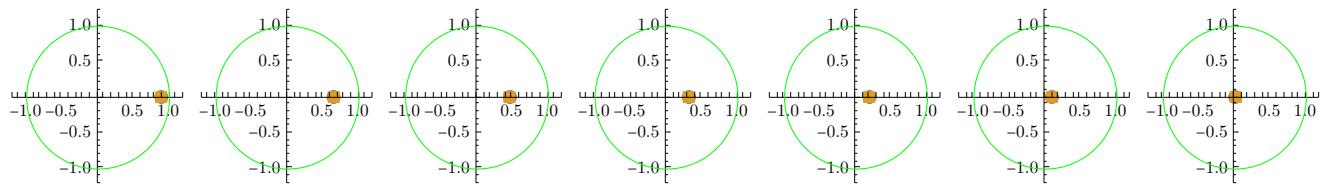
```
In[530]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

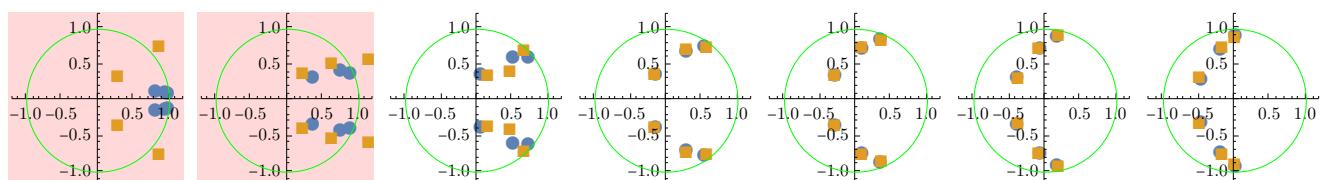
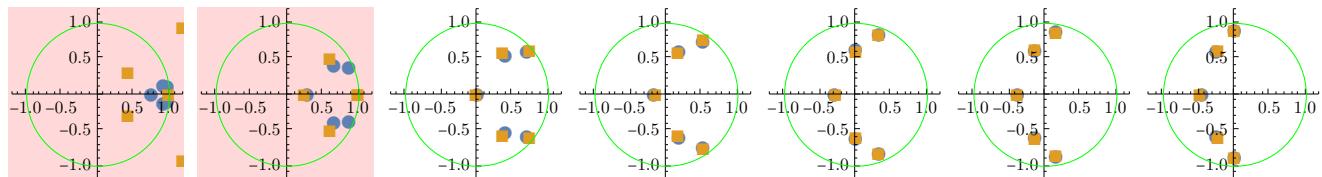
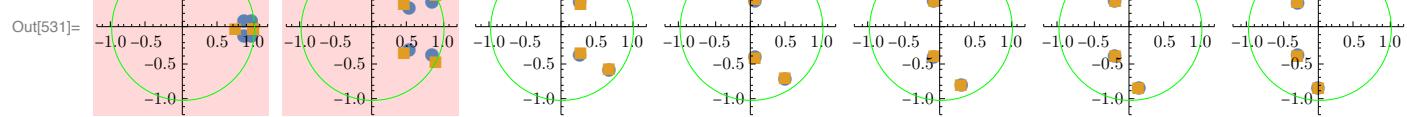
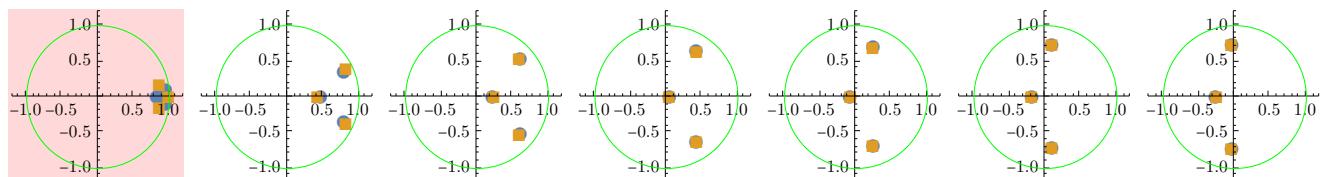
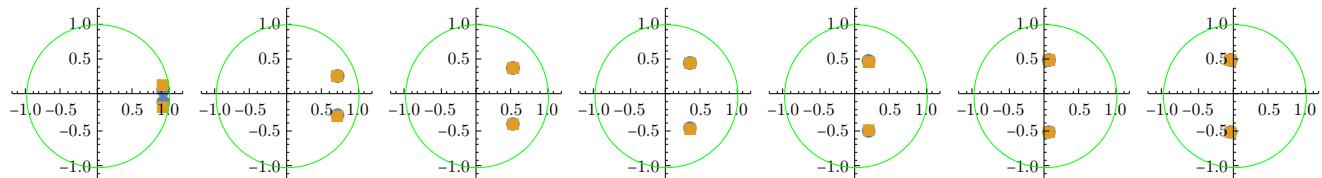


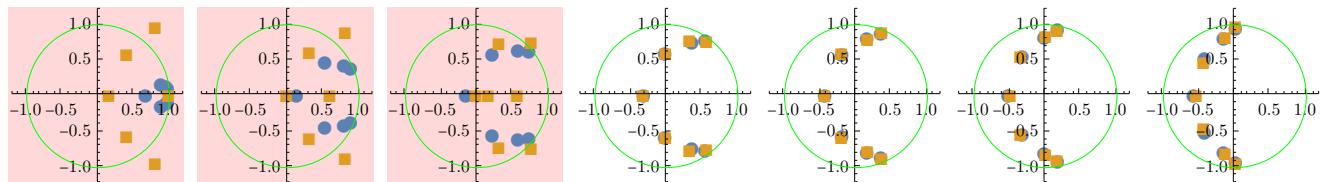


★ Chebyshev 2

```
In[531]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

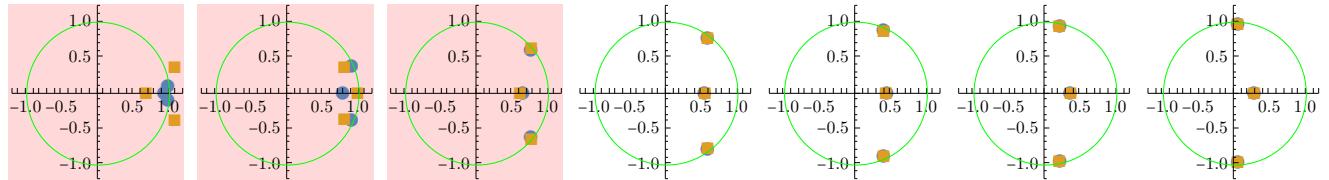
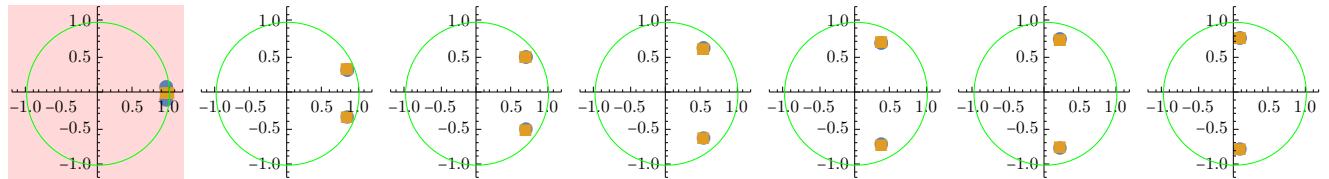
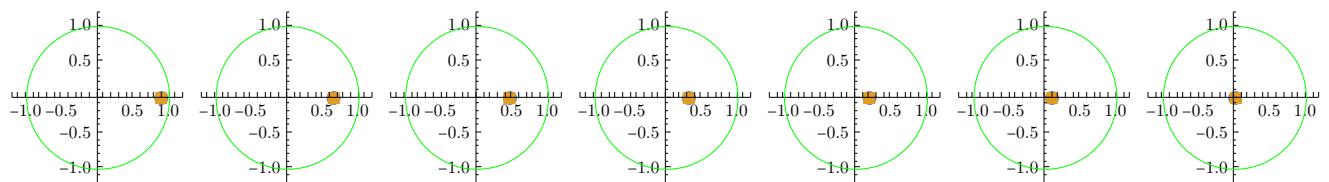


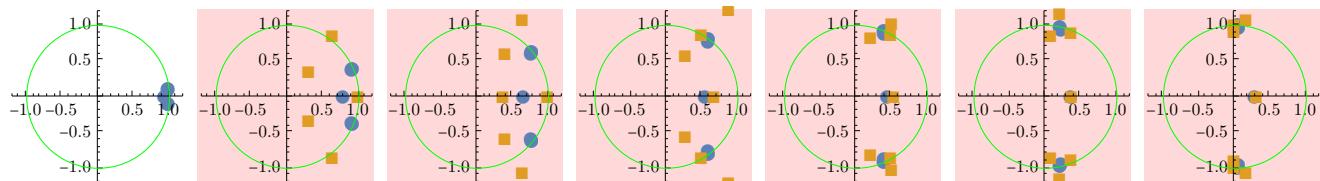
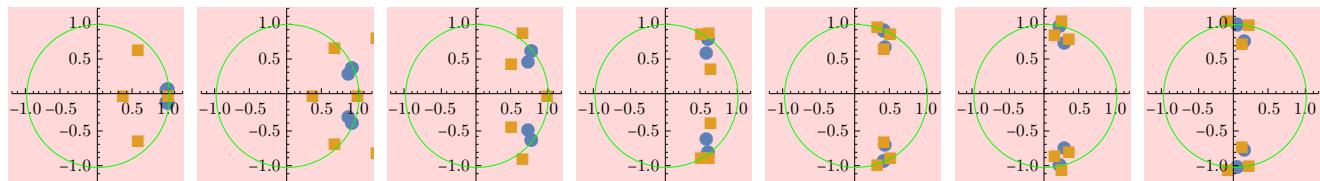
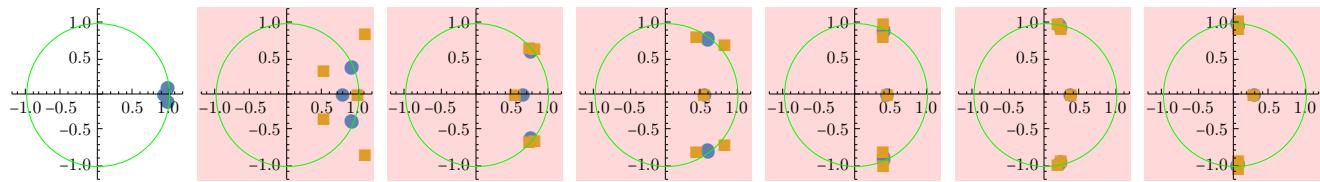
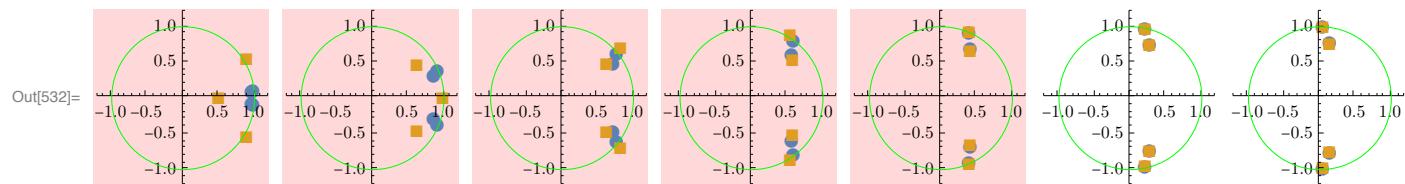




★ Eliptyczne

```
In[532]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





16.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

16.2.1 Dyskretyzacja

```
In[533]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[534]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[535]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

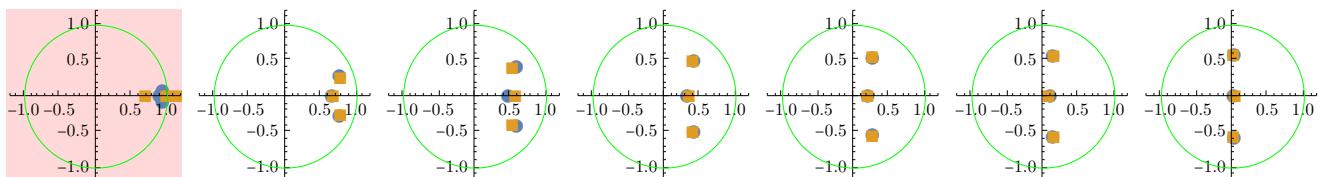
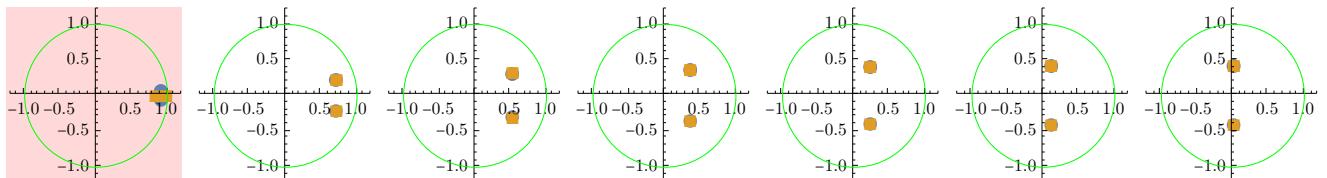
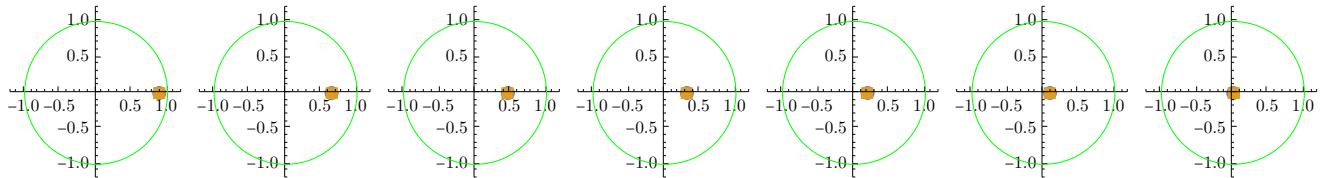
```
In[536]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

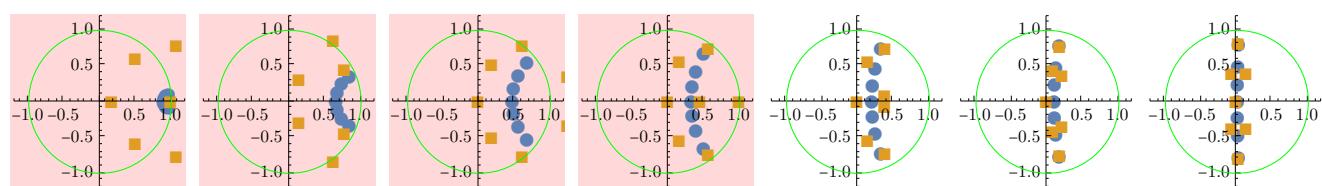
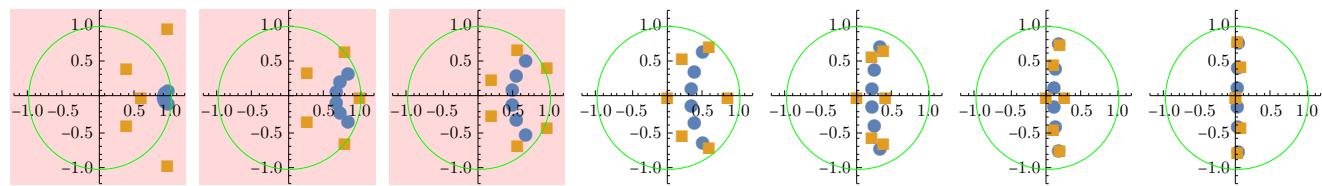
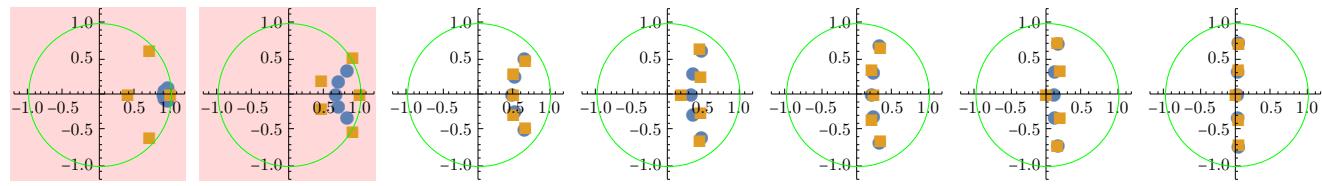
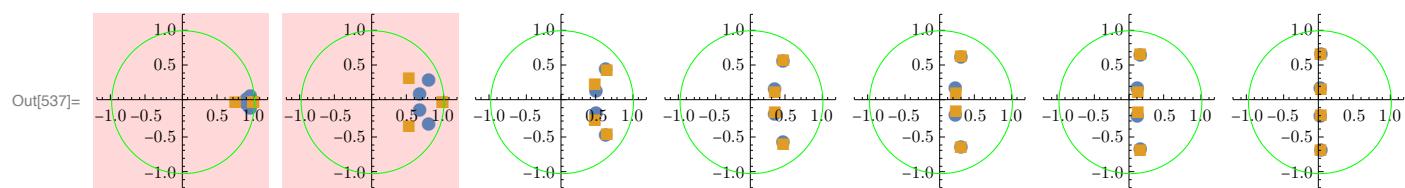
16.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

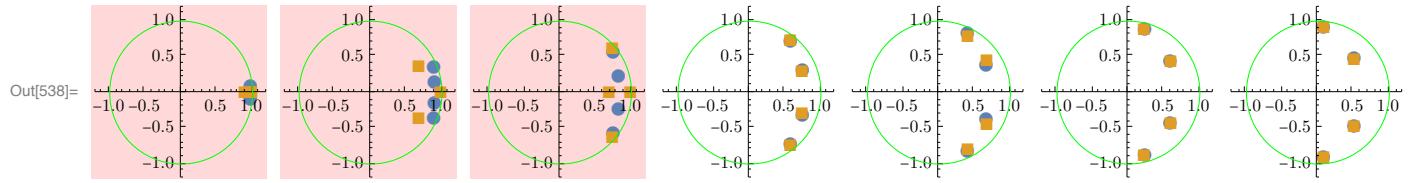
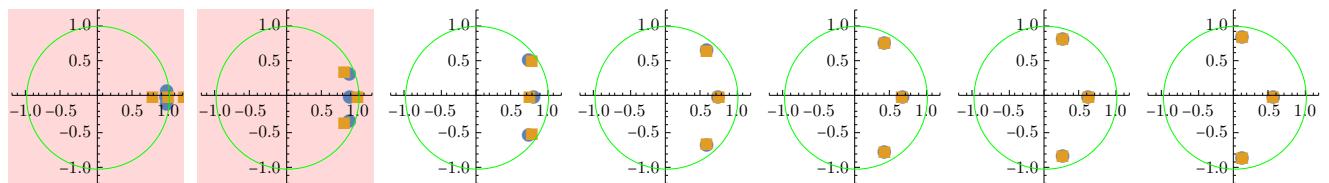
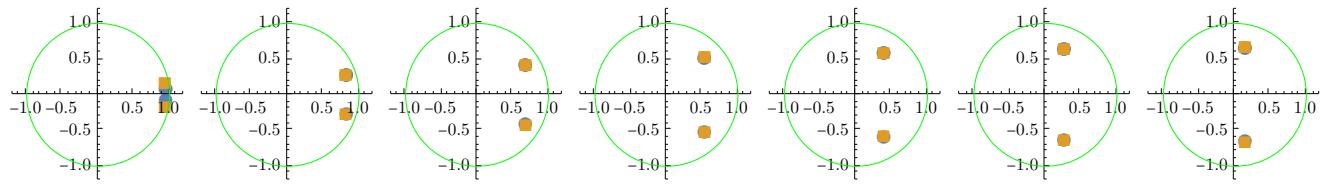
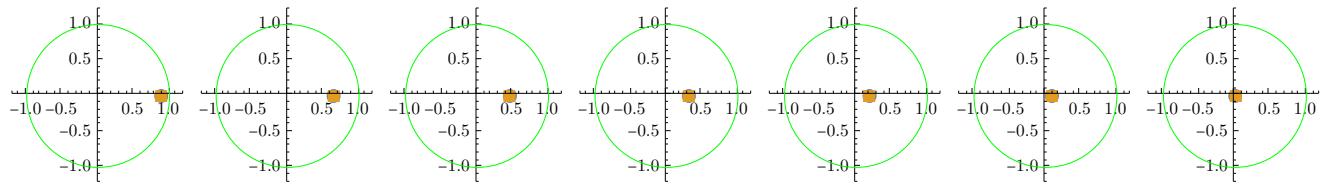
```
In[537]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

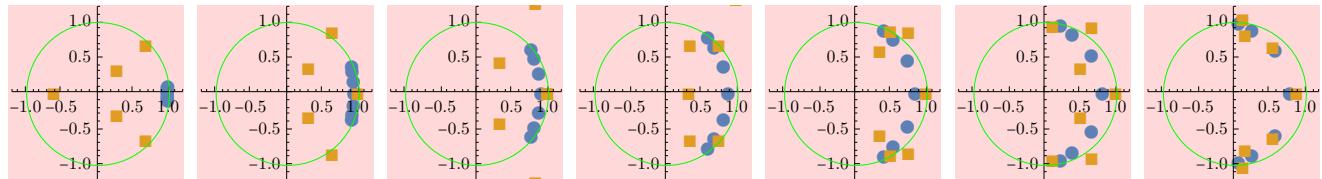
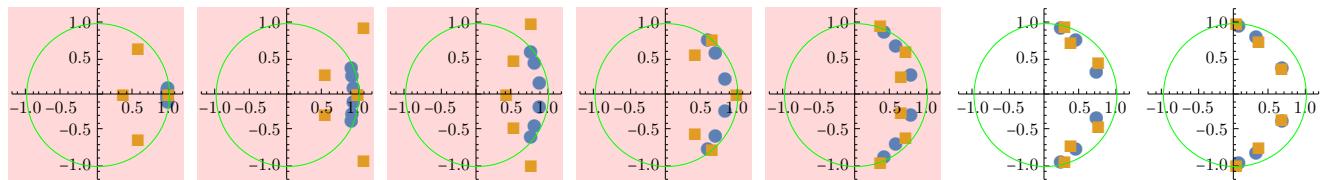
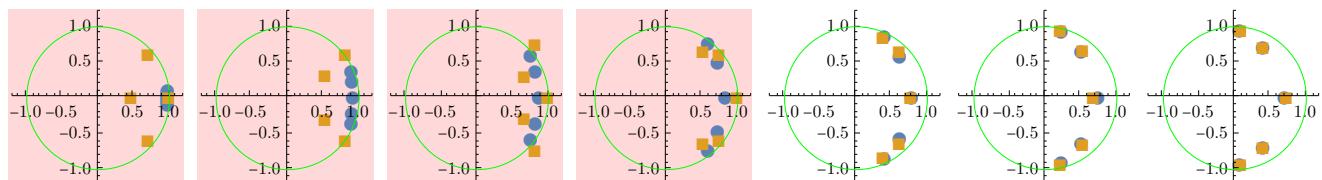




★ Chebyshev 1

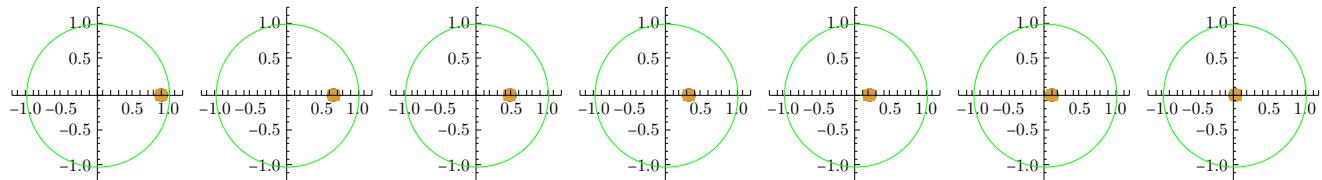
```
In[538]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

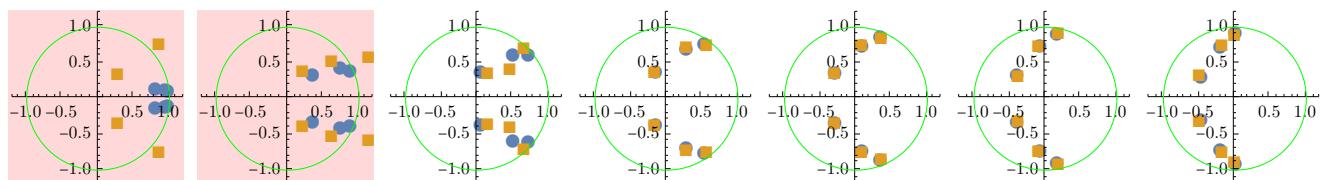
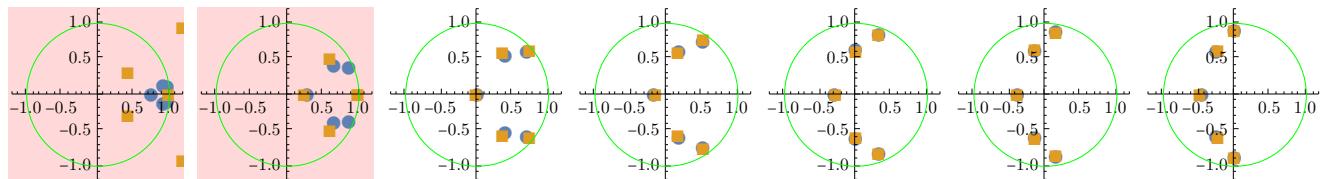
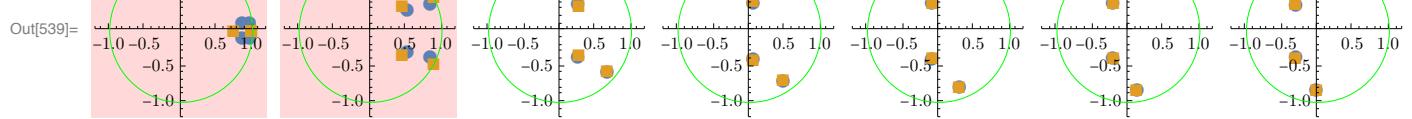
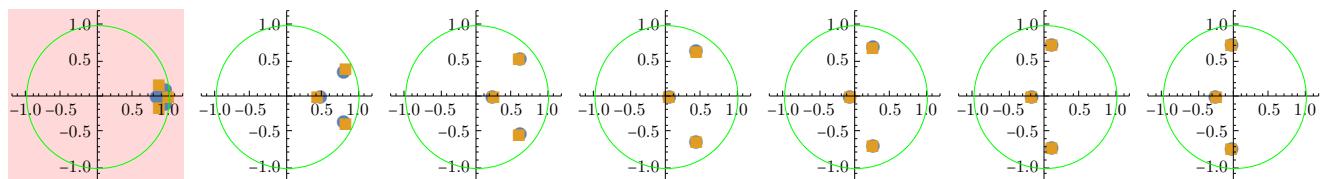
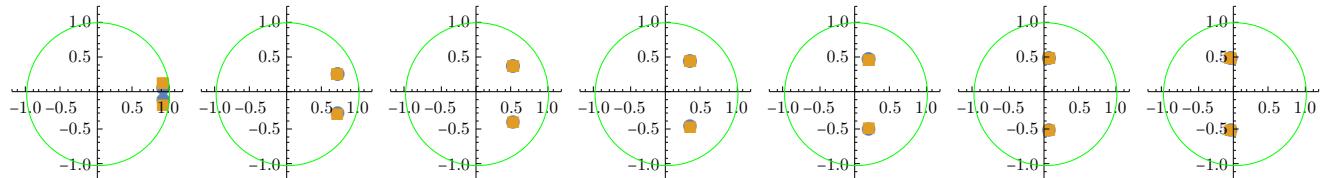


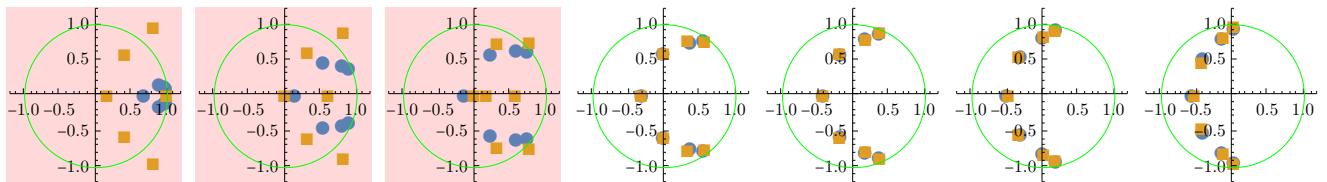


★ Chebyshev 2

```
In[539]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

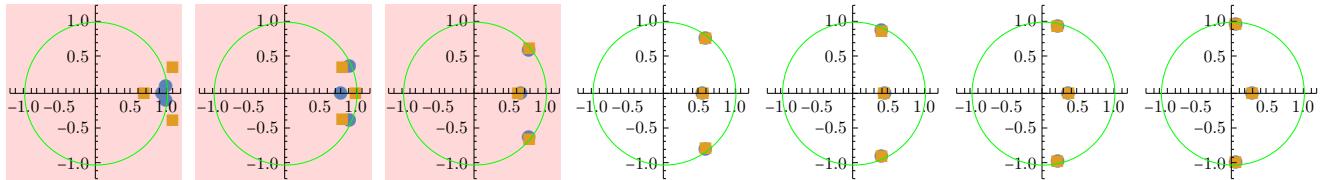
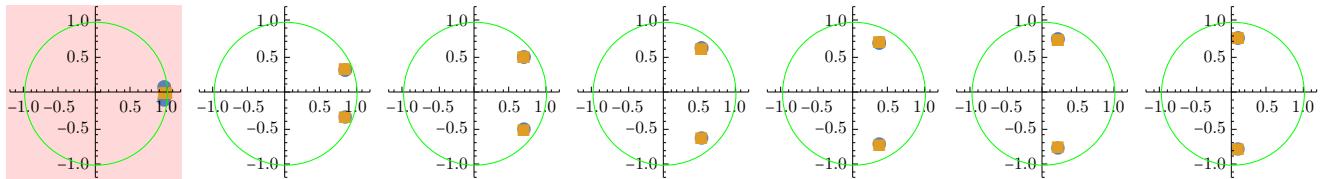
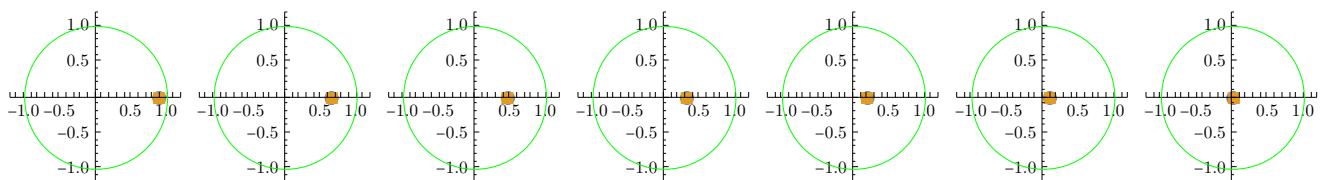


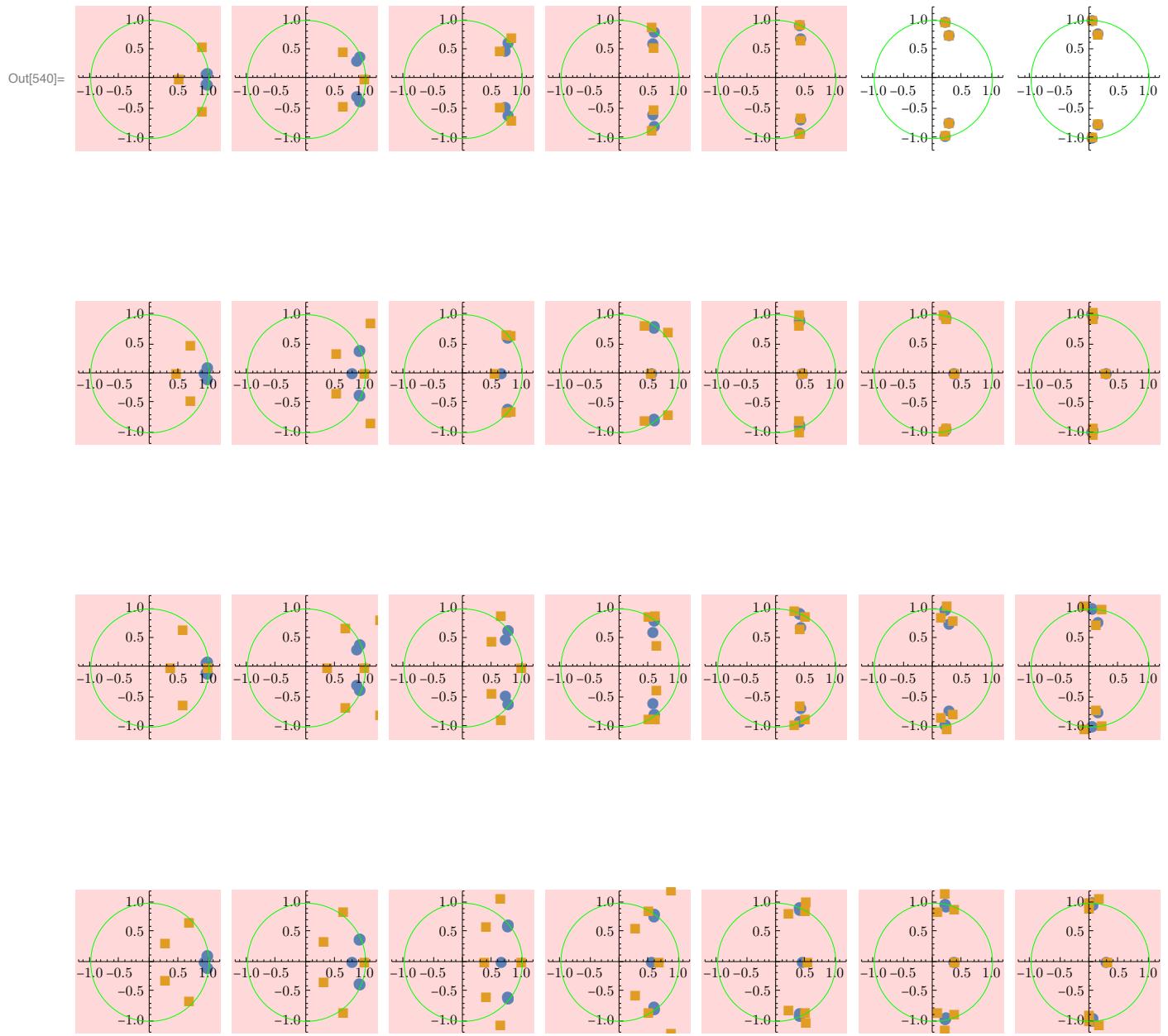




★ Eliptyczne

```
In[540]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





17 Badanie stabilności filtrów cyfrowych (8 bit)

In[541]:= **bity = 8;**

17.1 Dyskretyzacja na poziomie współczynników

17.1.1 Dyskretyzacja

In[542]:= **DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];**

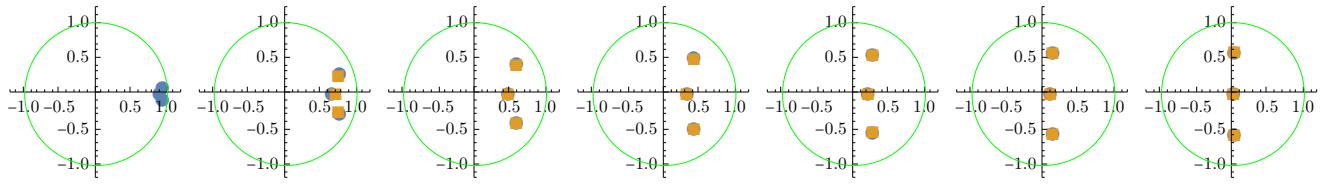
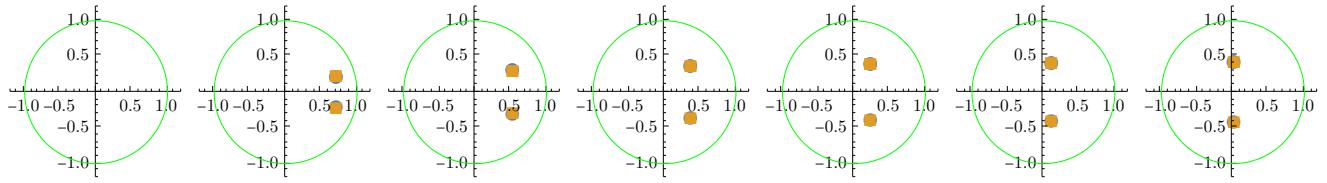
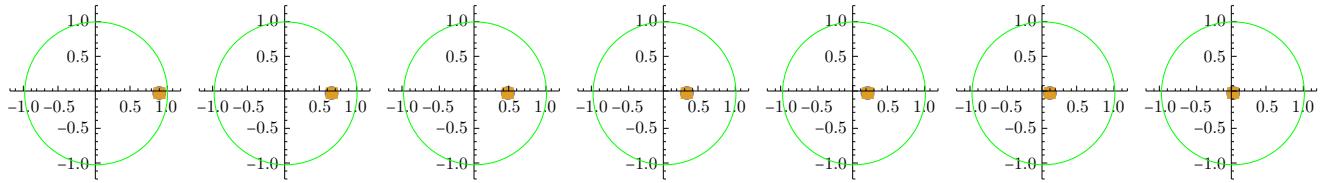
```
In[543]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[544]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[545]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

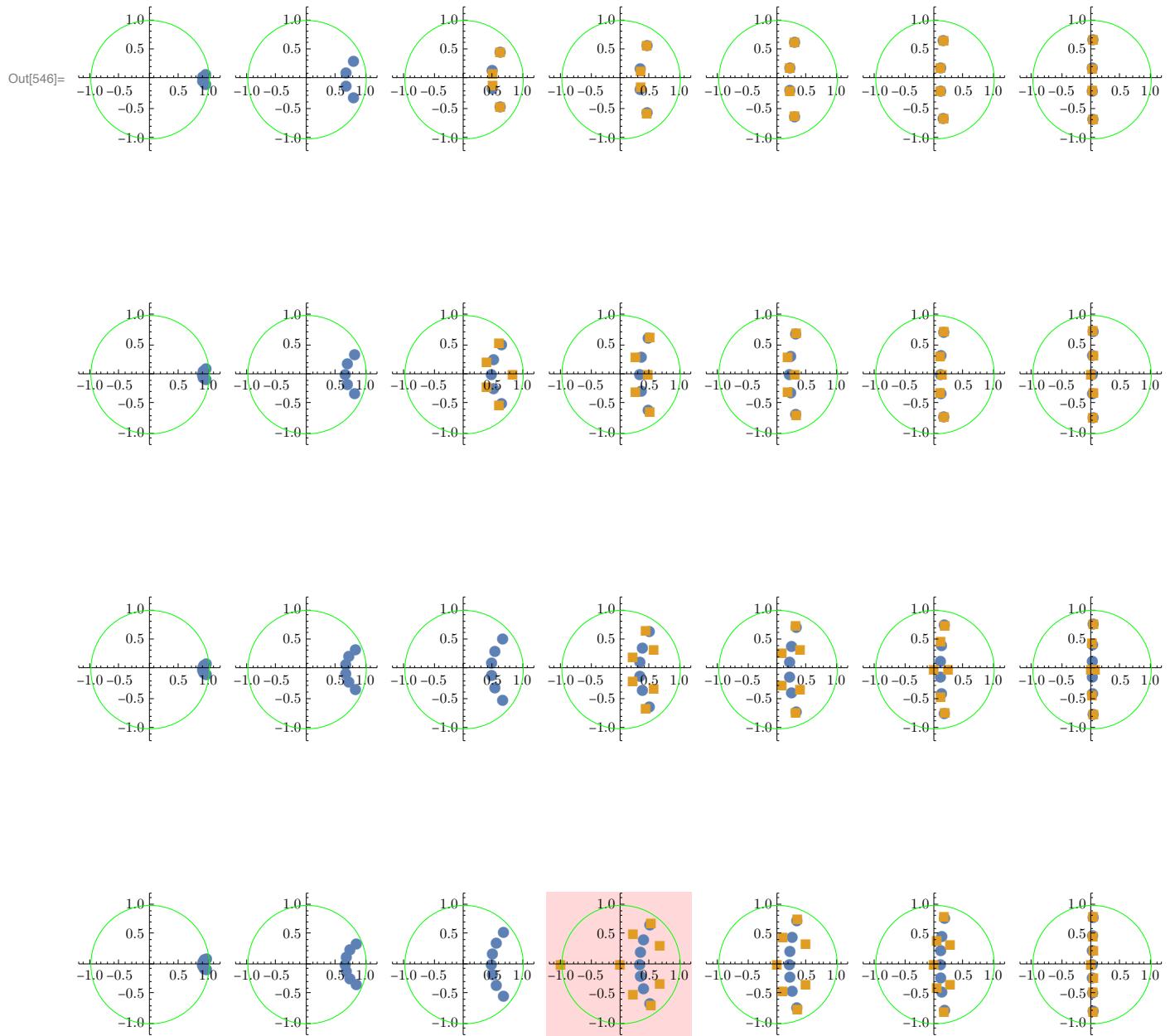
17.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

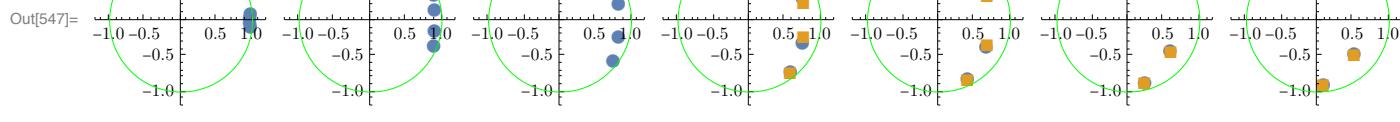
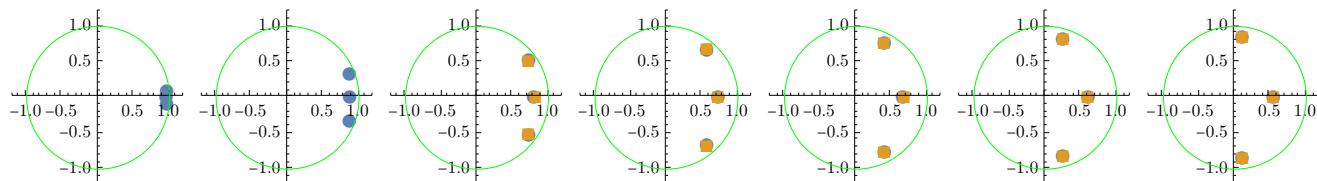
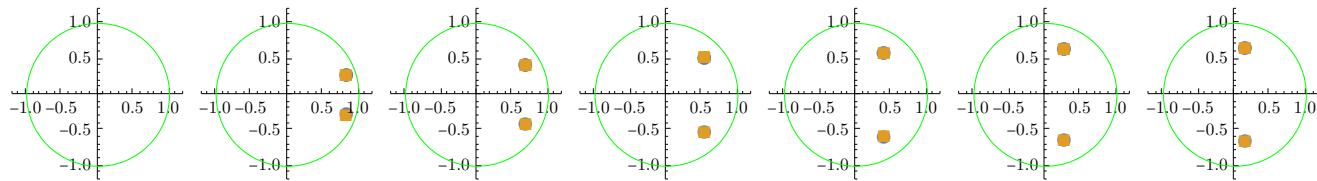
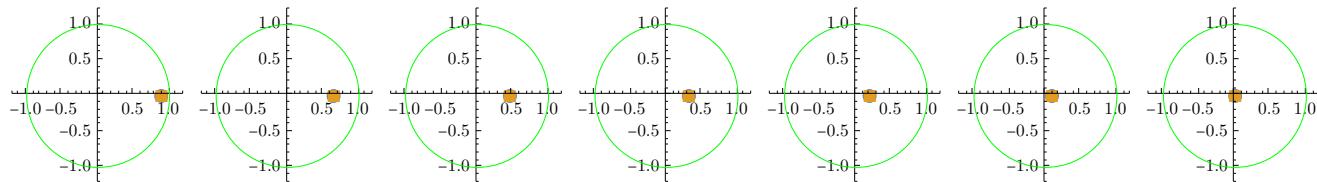
```
In[546]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

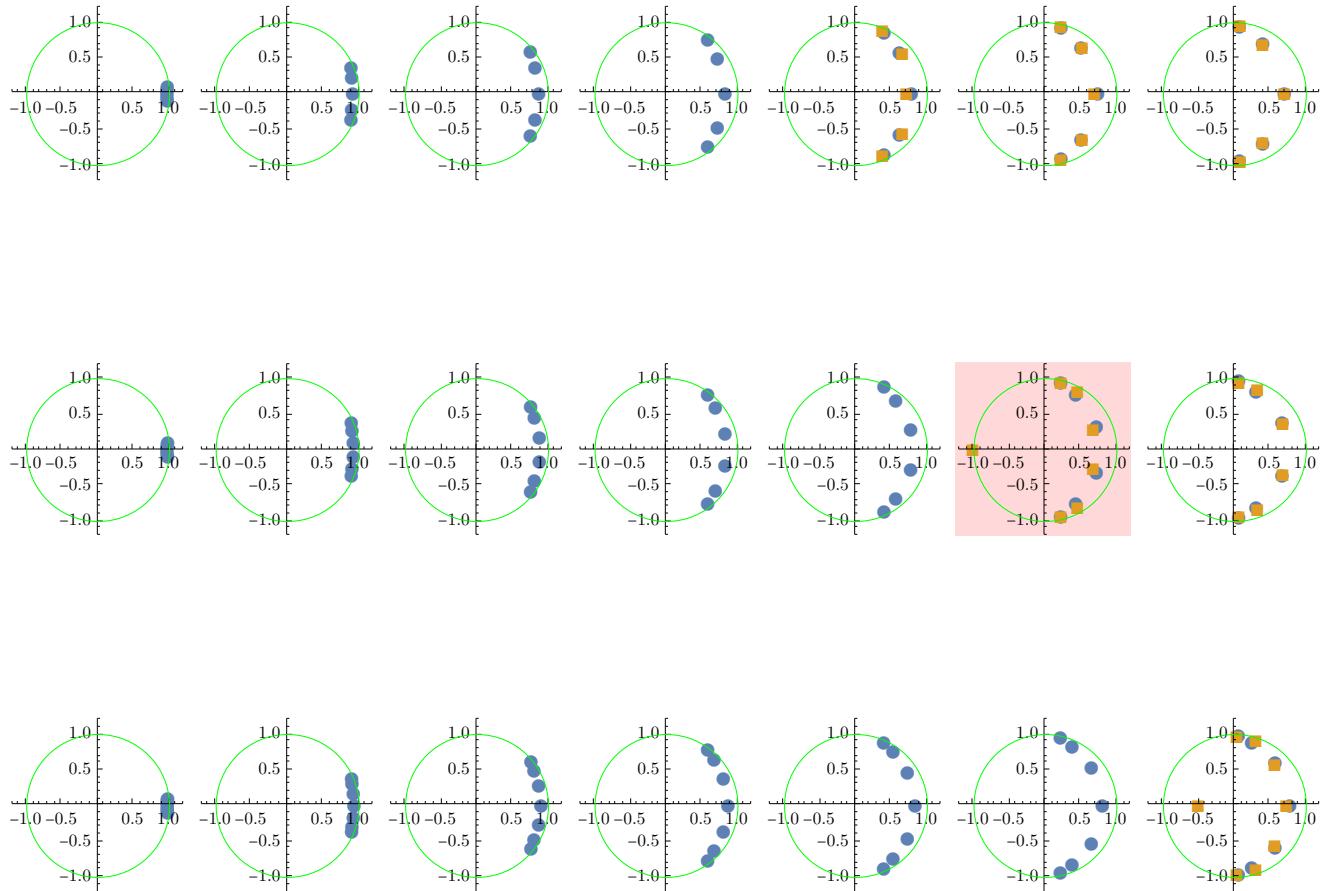




★ Chebyshev 1

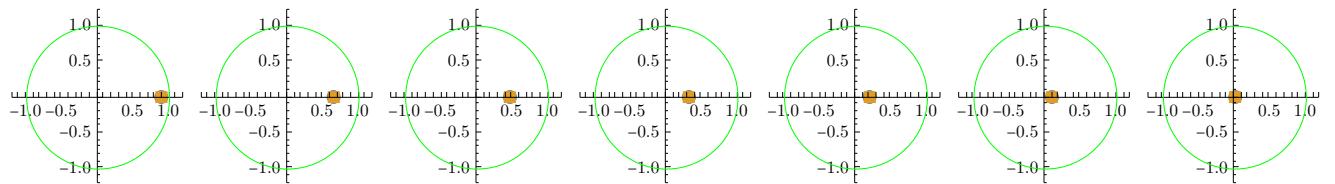
```
In[547]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

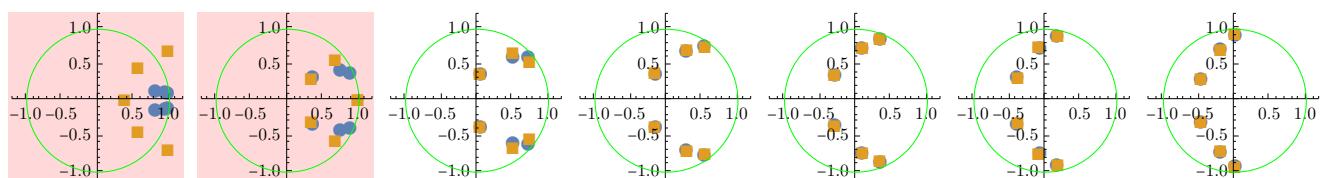
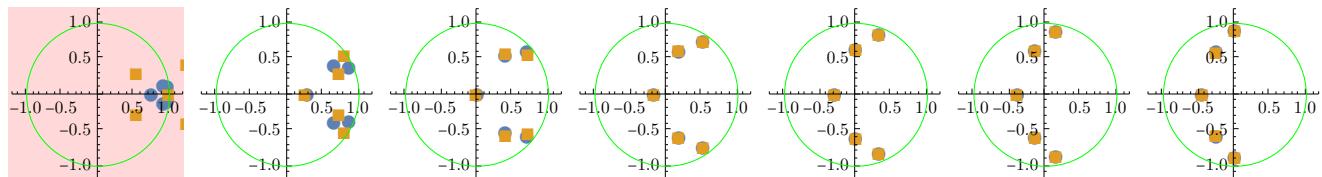
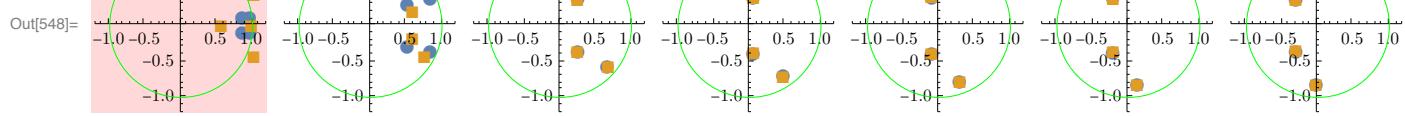
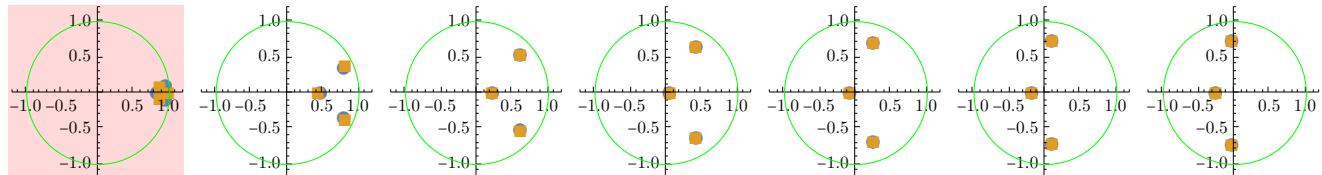
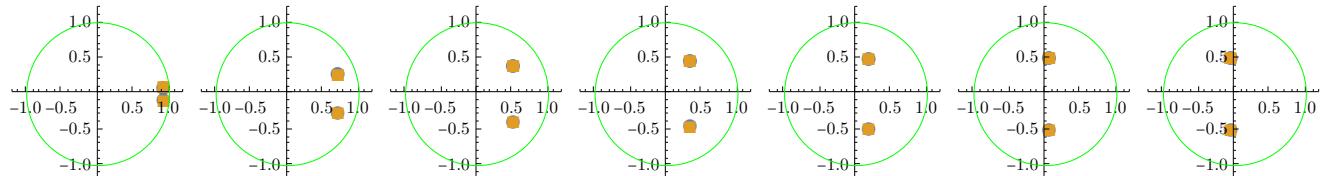


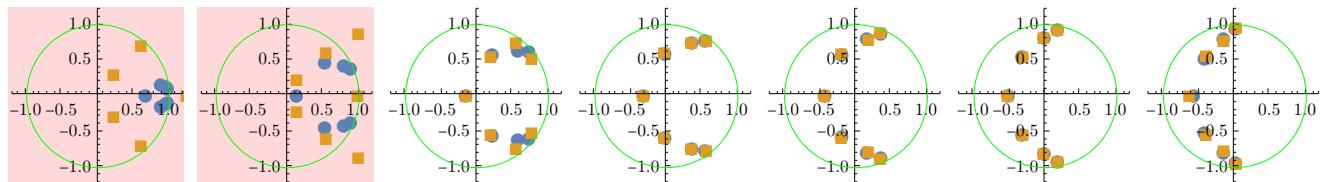


★ Chebyshev 2

```
In[548]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

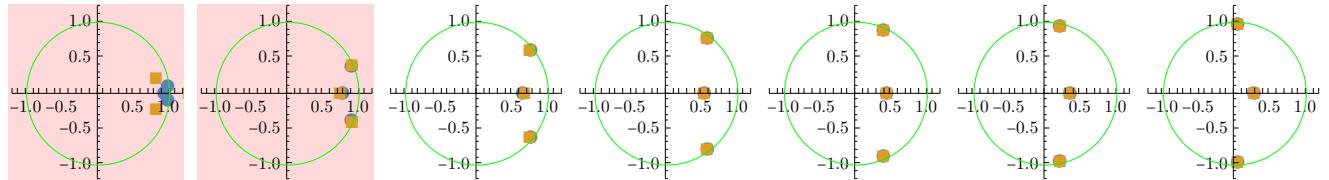
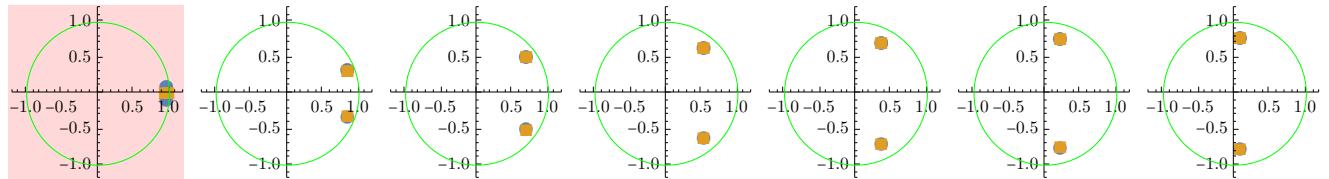
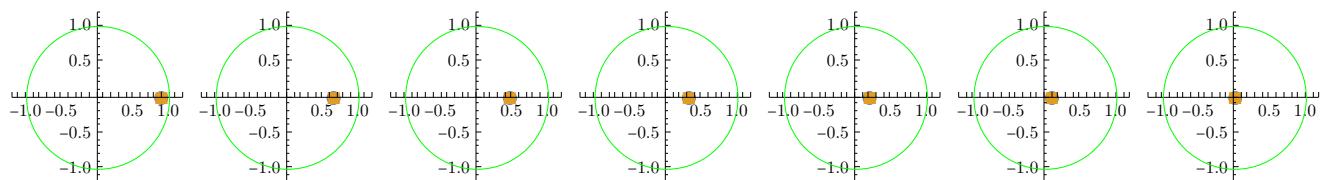


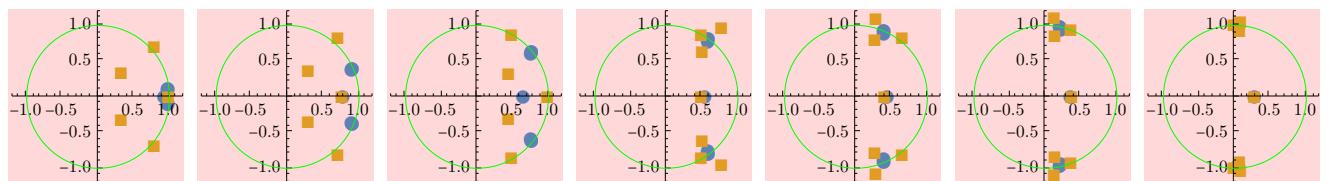
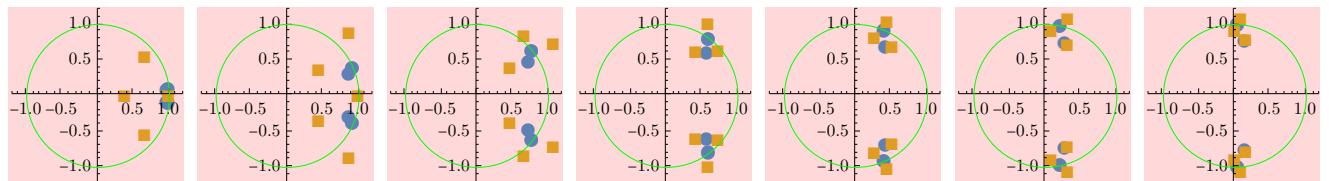
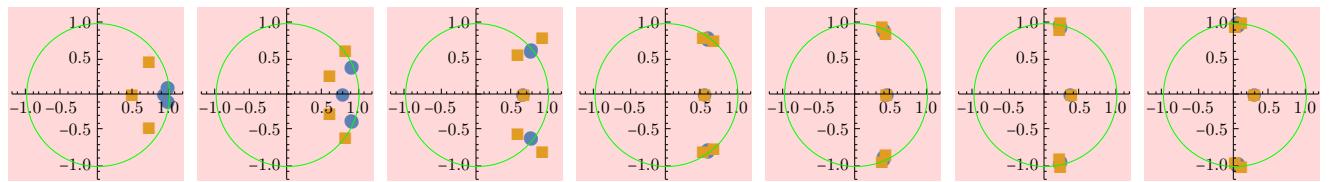
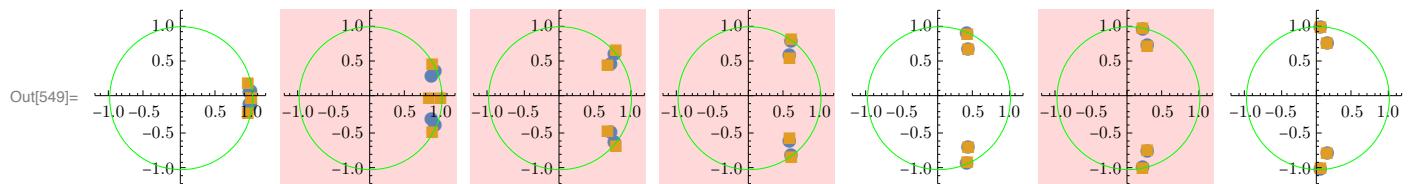




★ Eliptyczne

```
In[549]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





17.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

17.2.1 Dyskretyzacja

```
In[550]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[551]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[552]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

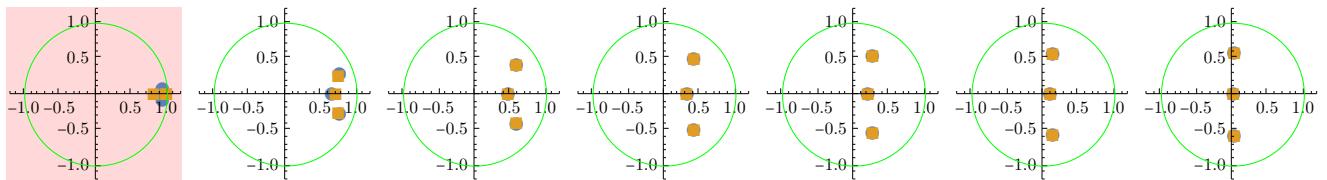
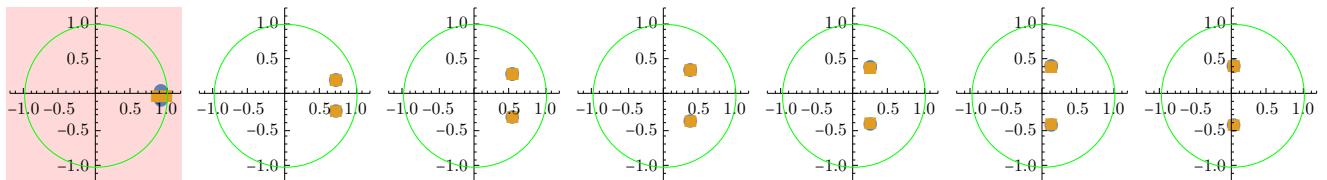
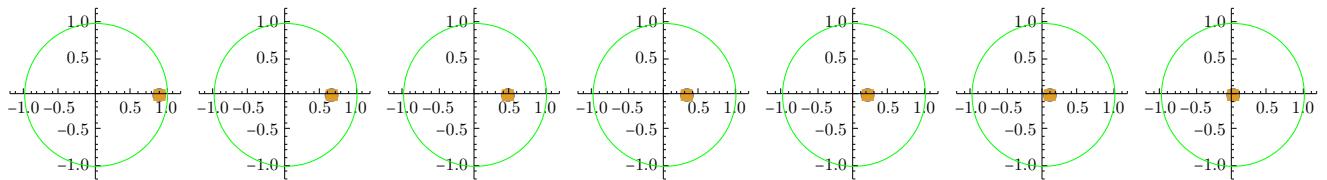
```
In[553]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

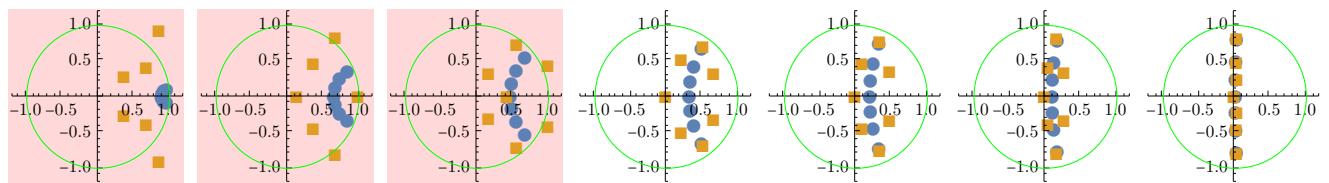
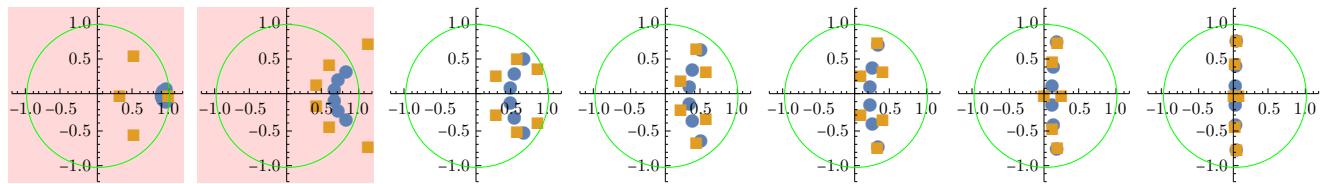
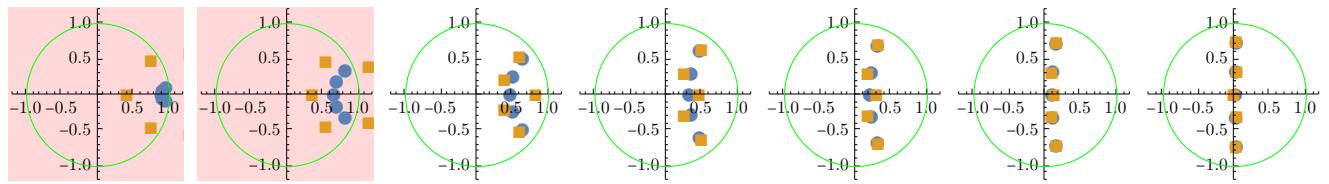
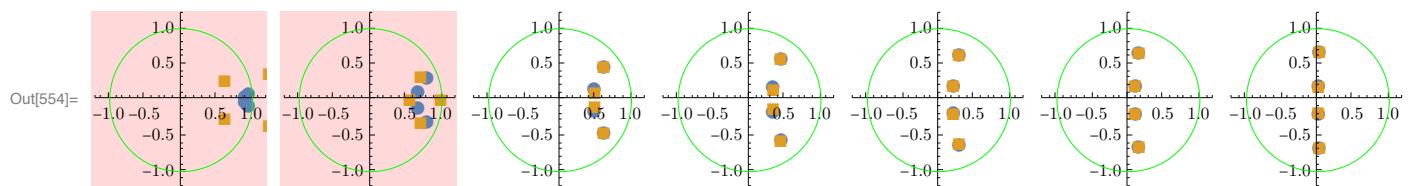
17.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

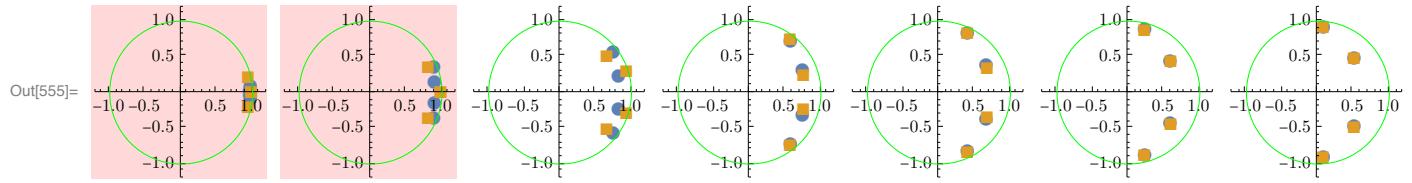
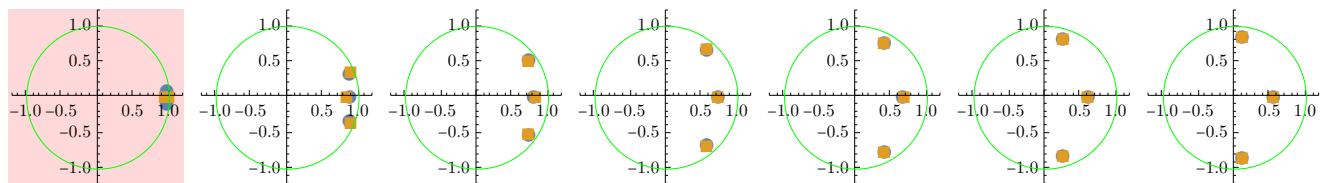
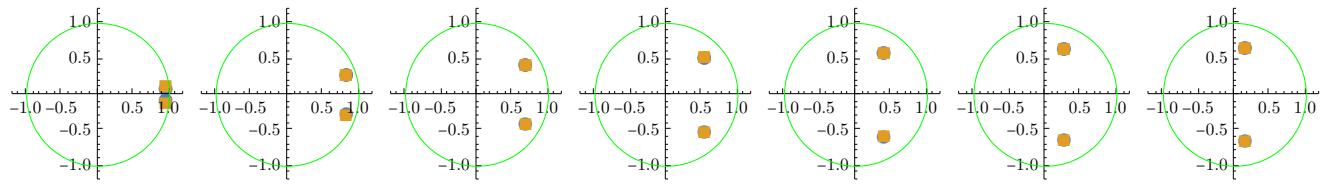
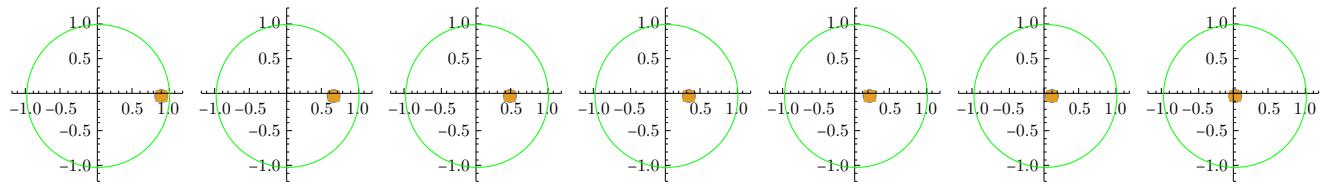
```
In[554]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

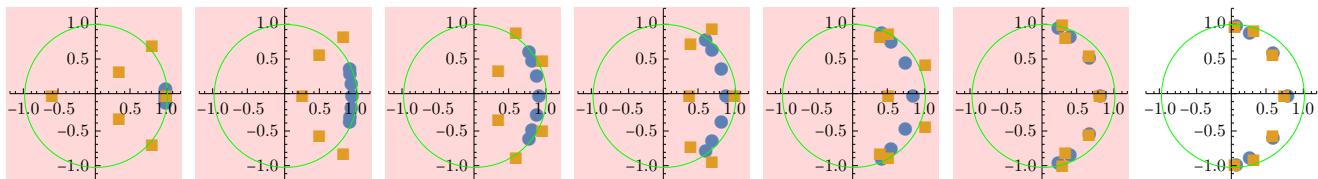
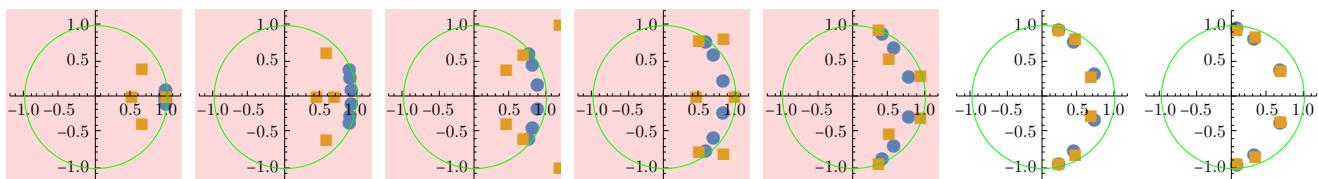
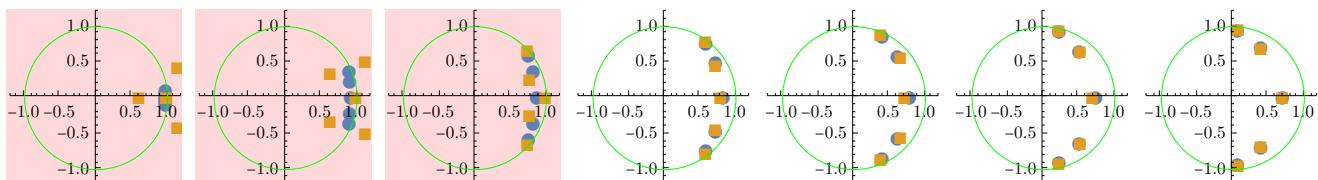




★ Chebyshev 1

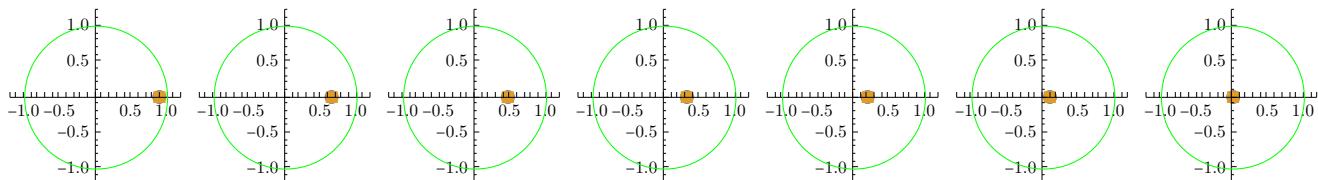
```
In[555]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

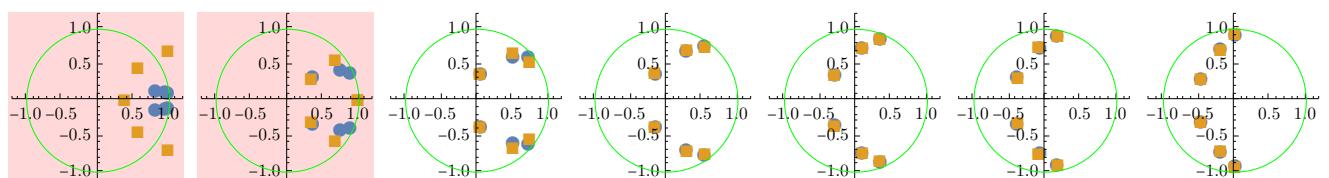
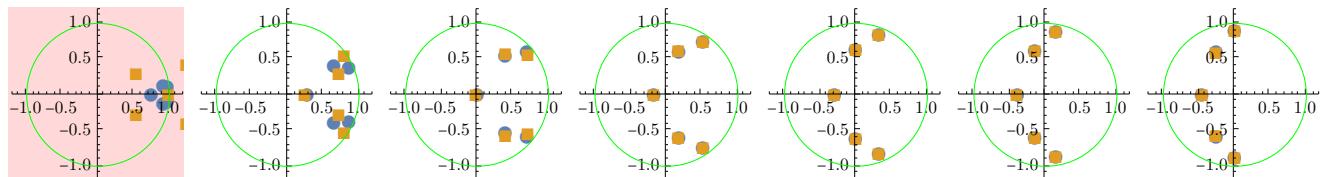
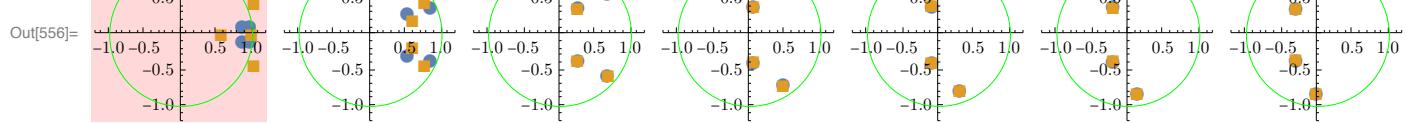
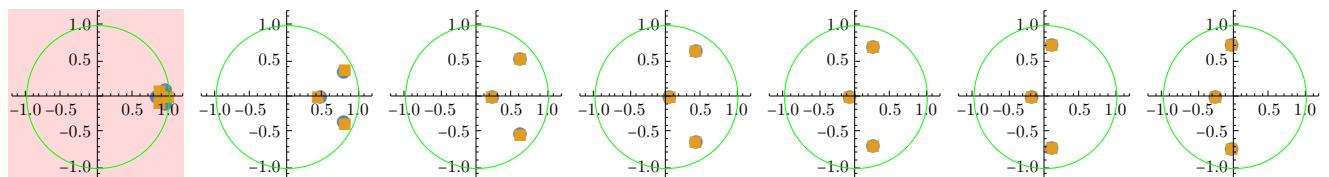
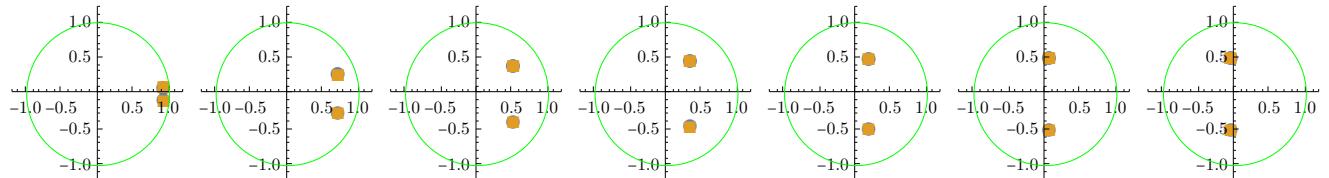


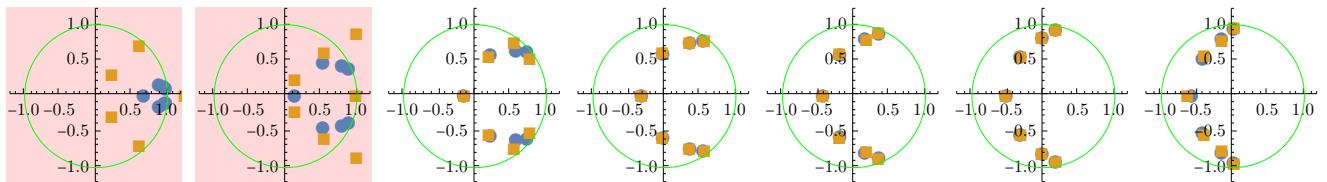


★ Chebyshev 2

```
In[556]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

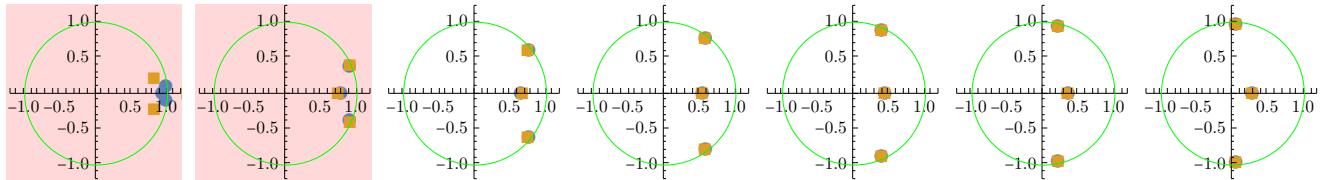
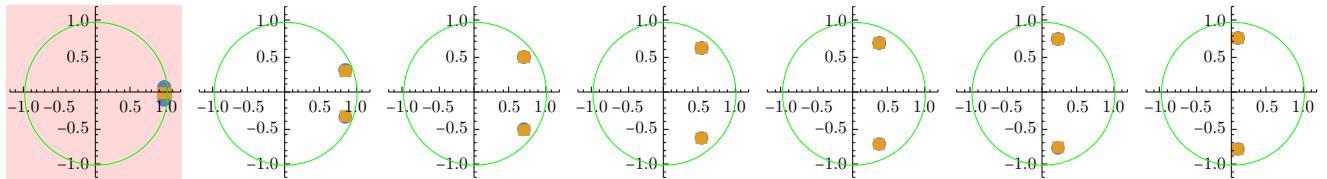
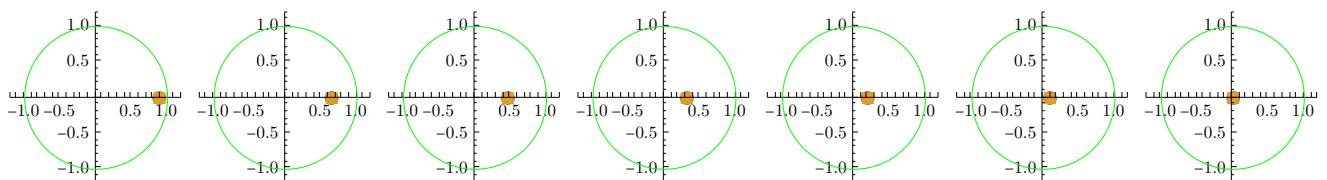


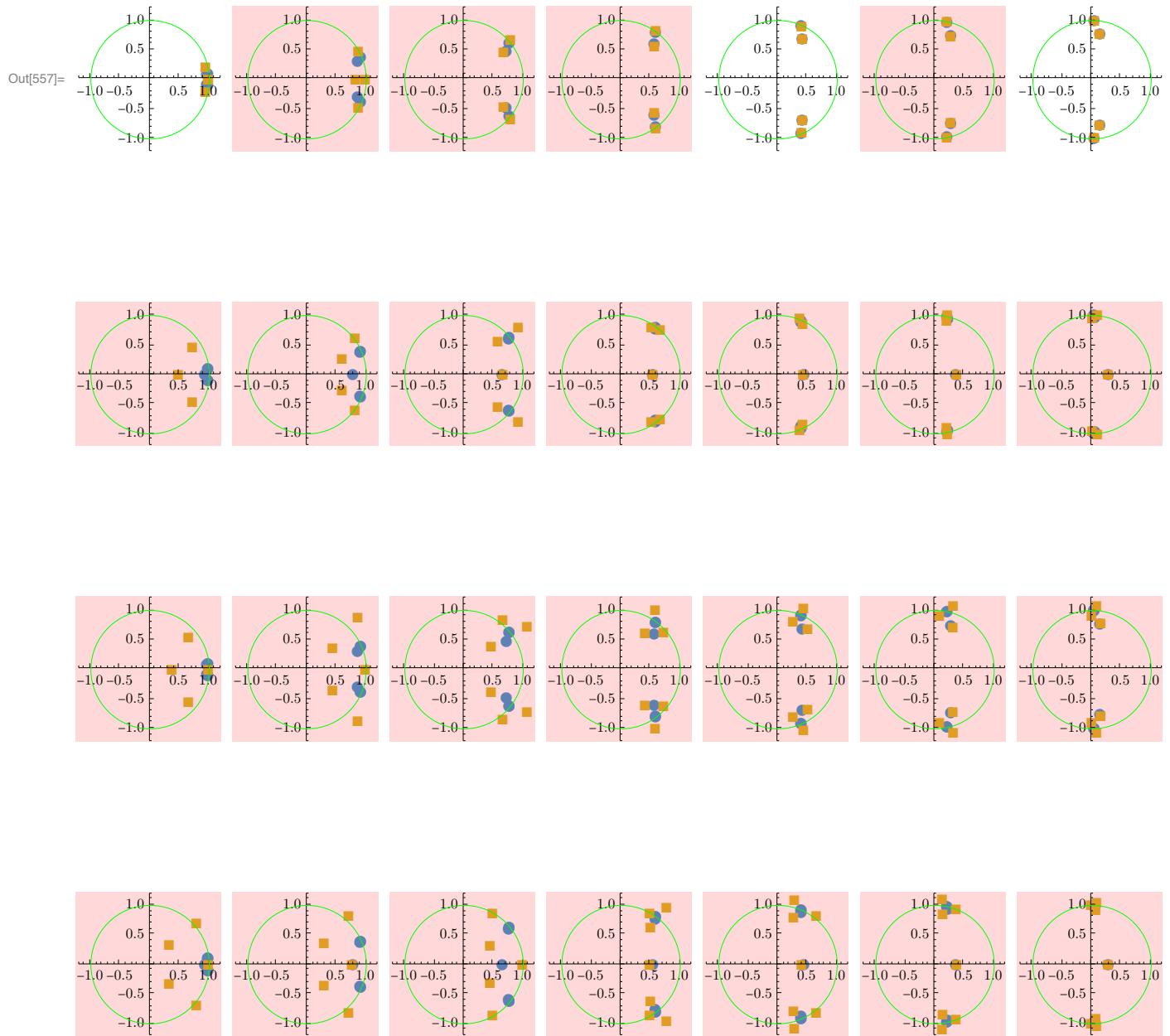




★ Eliptyczne

```
In[557]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





18 Badanie stabilności filtrów cyfrowych (9 bit)

In[558]:= **bity = 9;**

18.1 Dyskretyzacja na poziomie współczynników

18.1.1 Dyskretyzacja

In[559]:= **DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];**

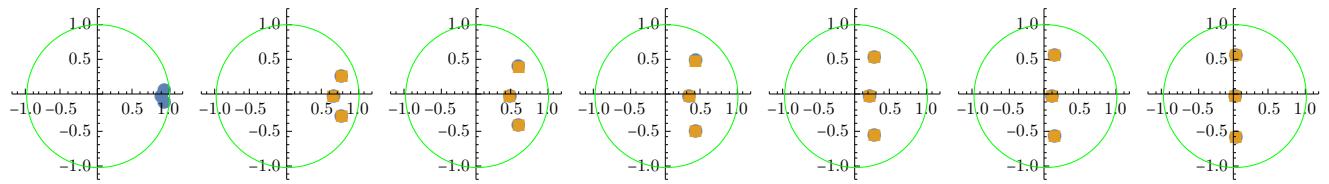
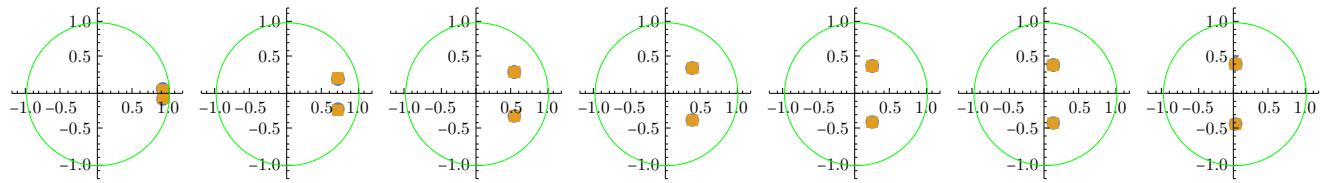
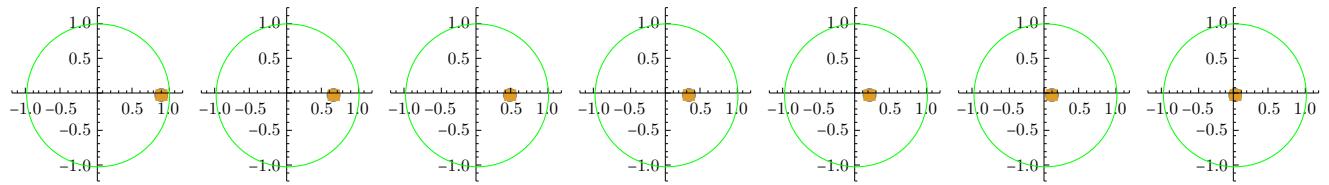
```
In[560]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[561]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[562]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

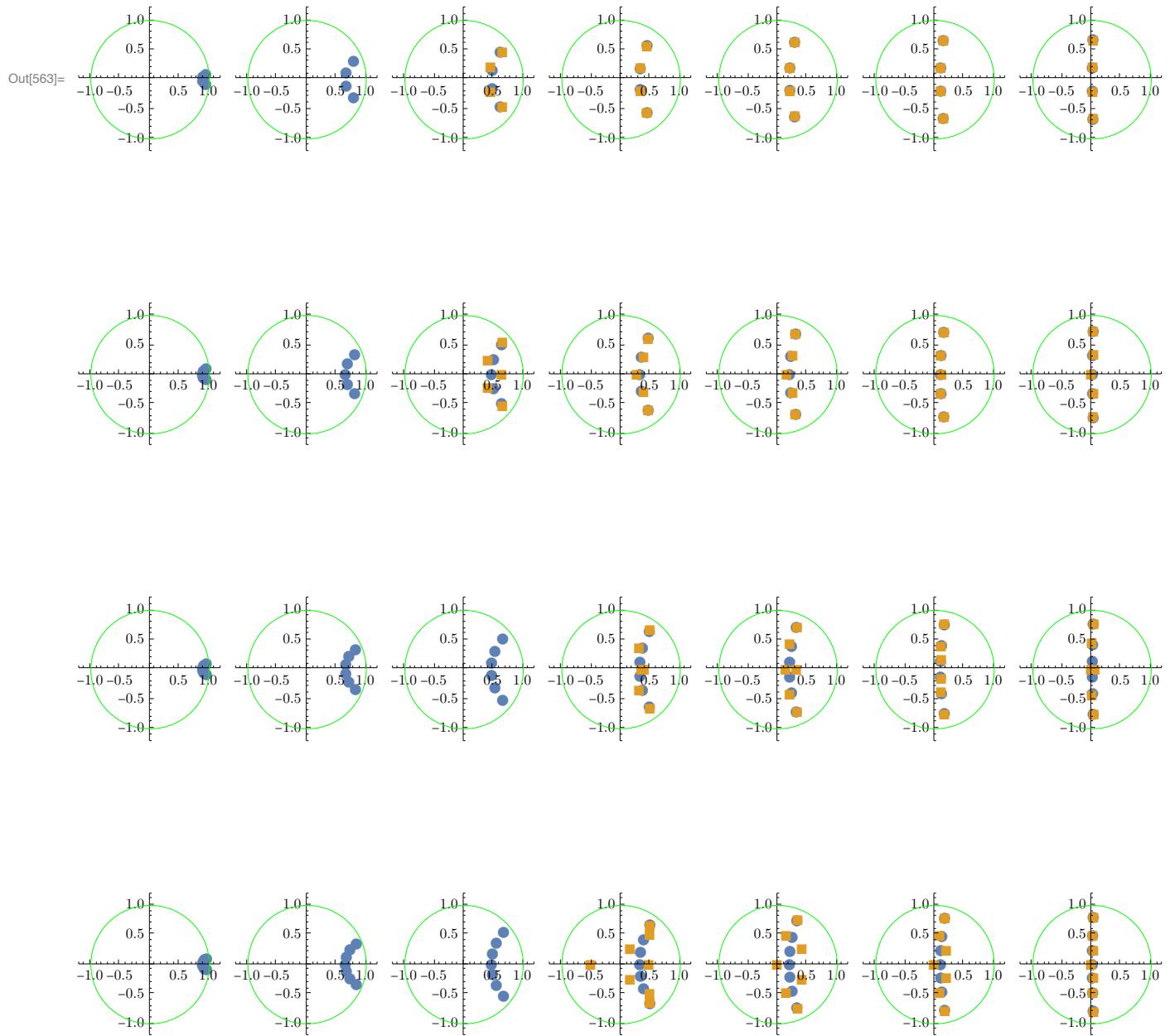
18.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

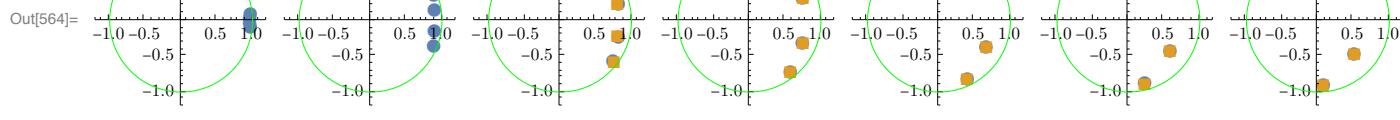
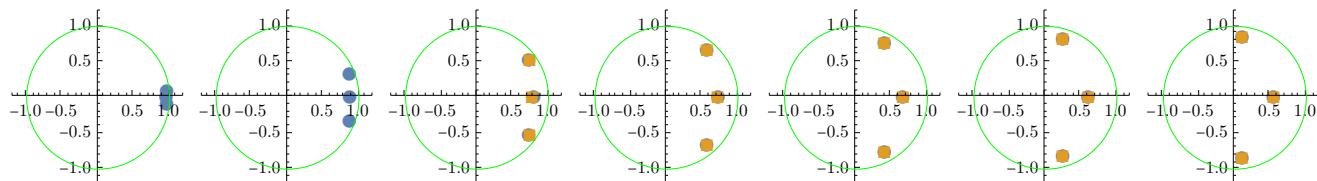
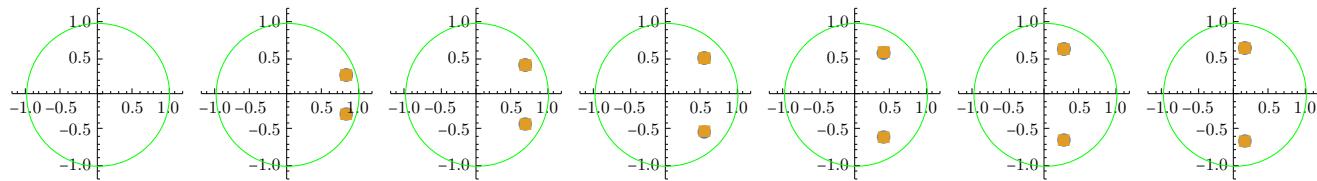
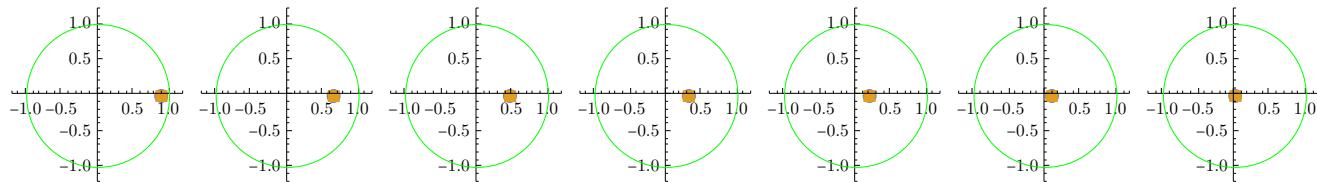
```
In[563]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

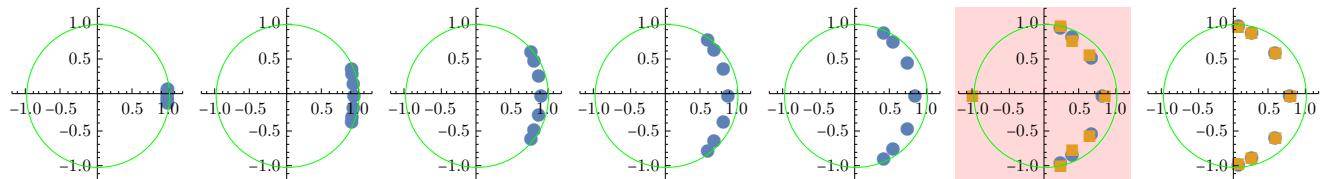
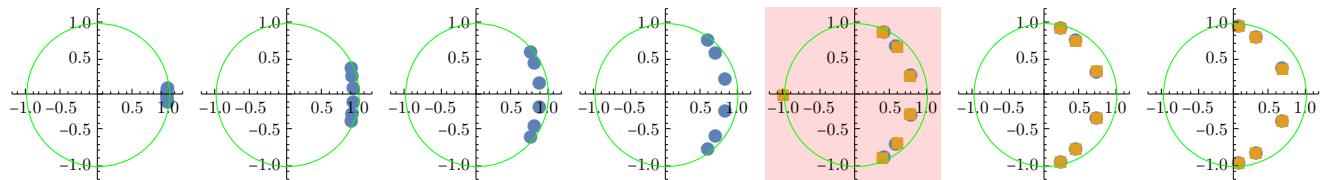
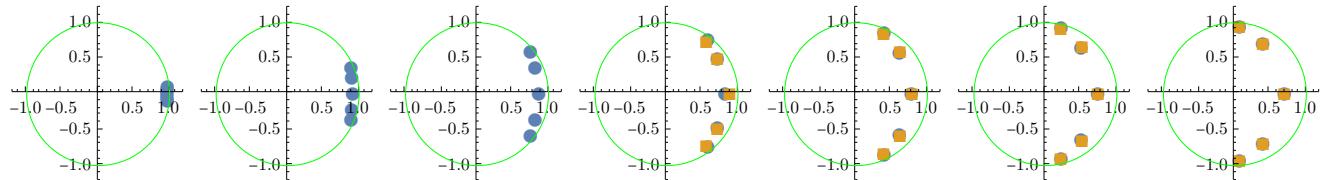




★ Chebyshev 1

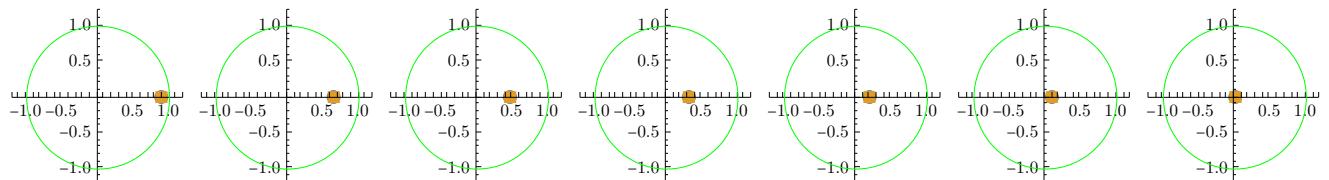
```
In[564]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

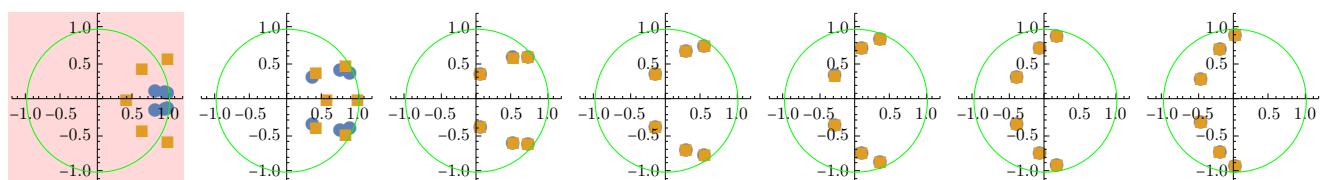
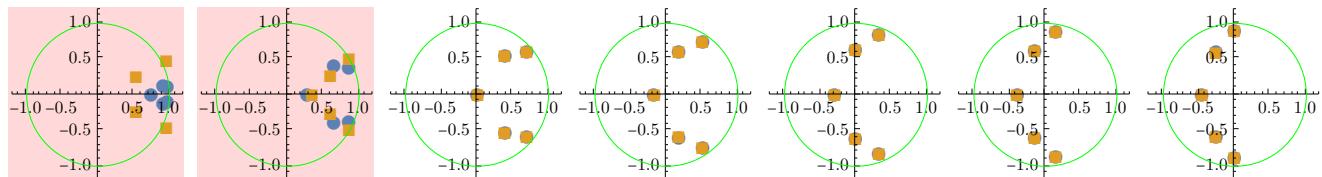
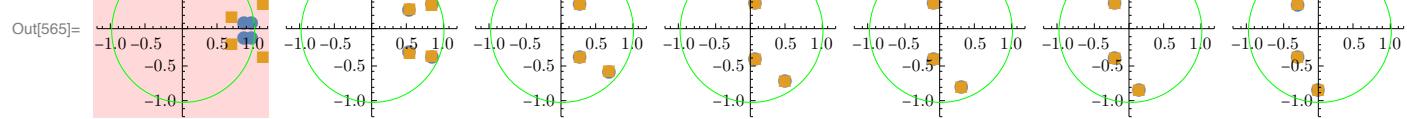
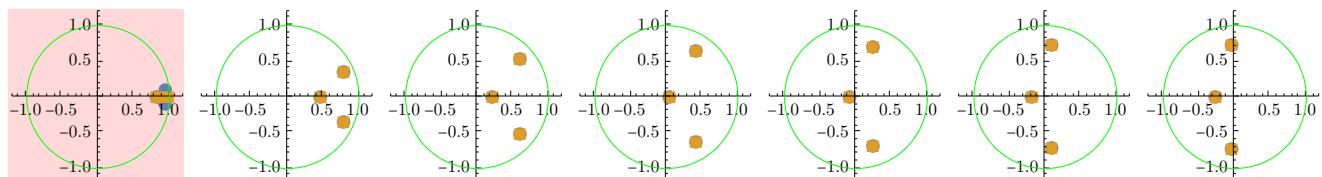
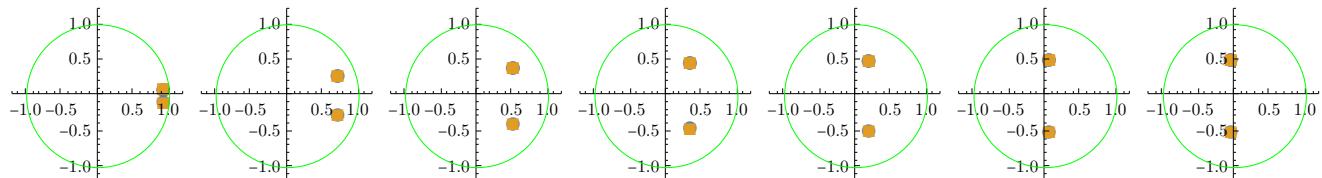


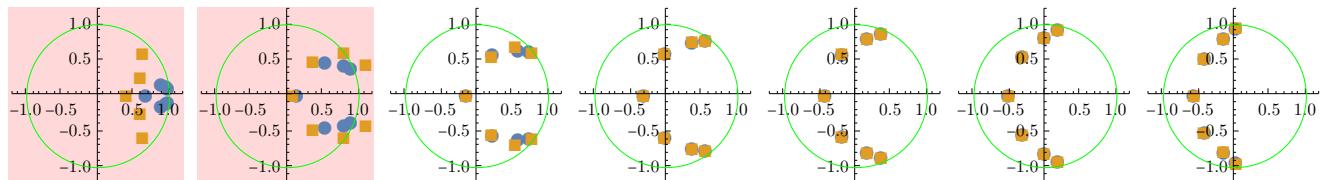


★ Chebyshev 2

```
In[565]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

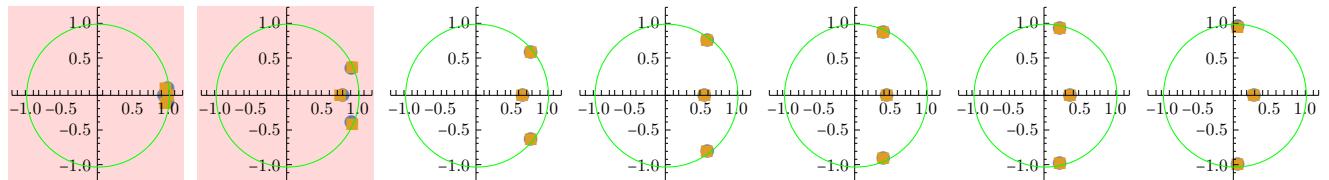
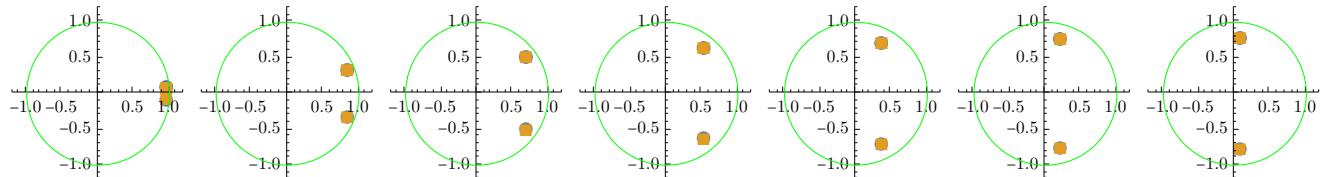
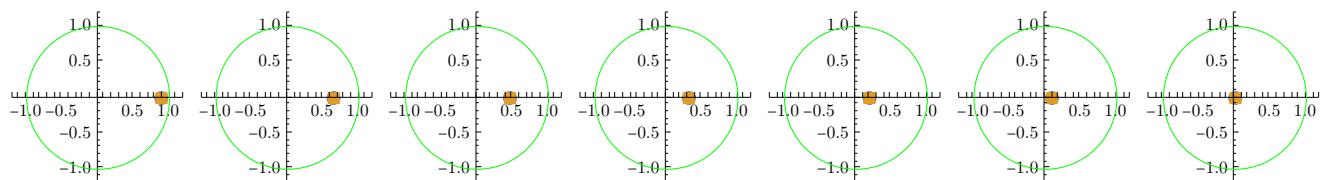


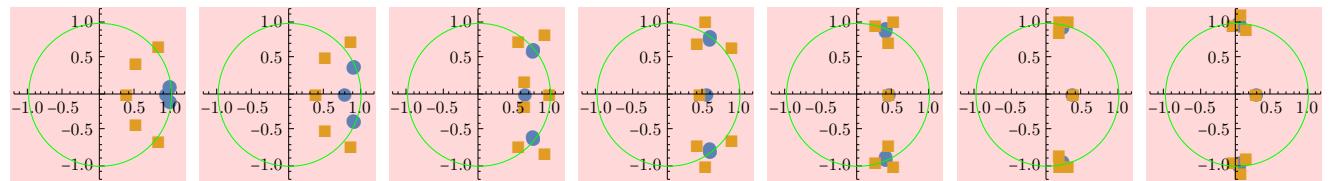
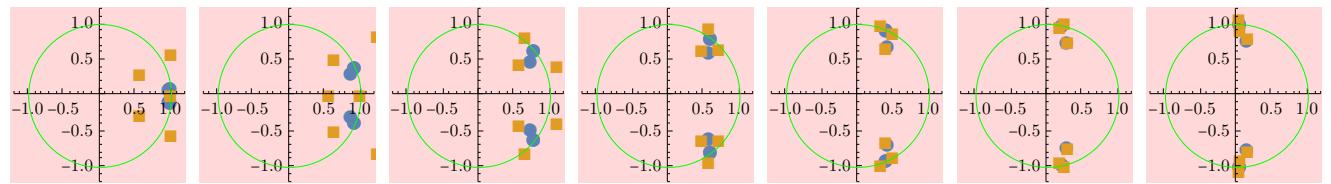
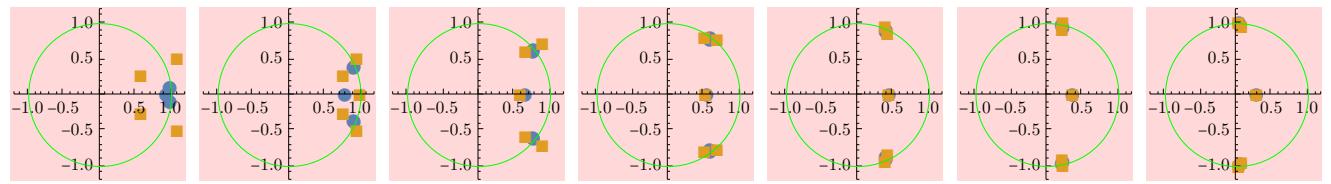
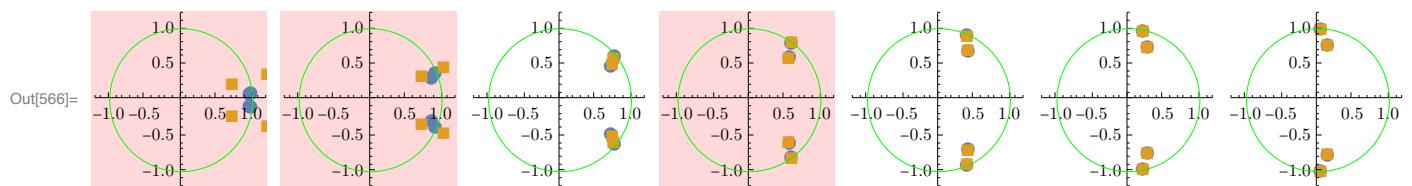




★ Eliptyczne

```
In[566]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





18.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

18.2.1 Dyskretyzacja

```
In[567]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[568]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[569]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

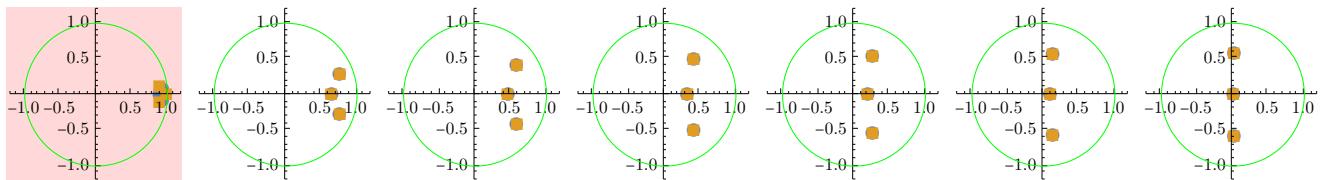
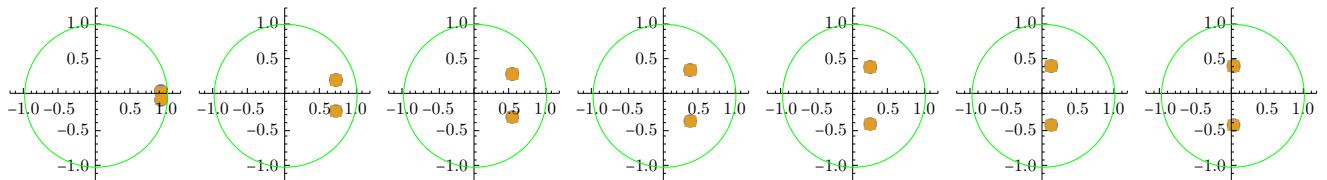
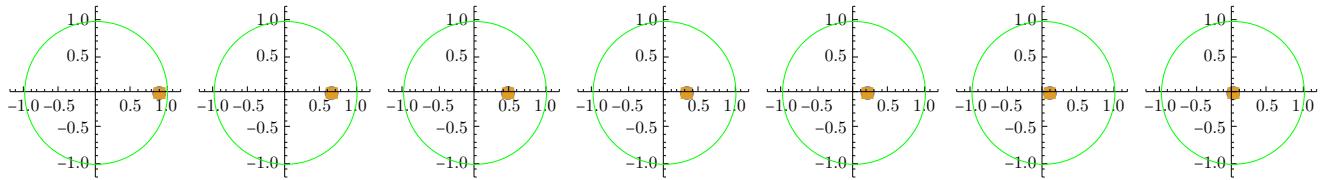
```
In[570]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

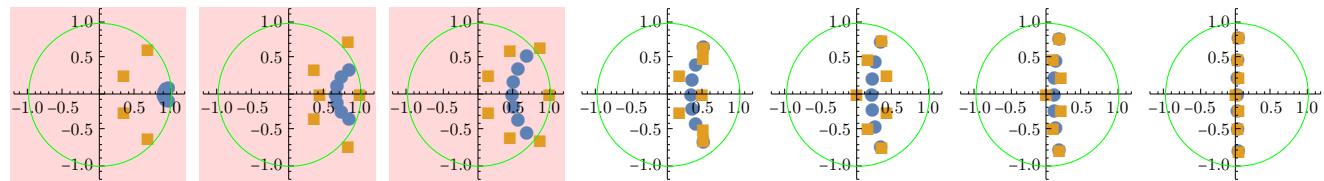
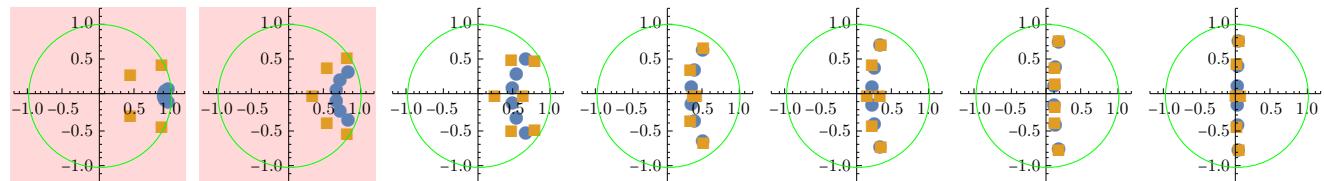
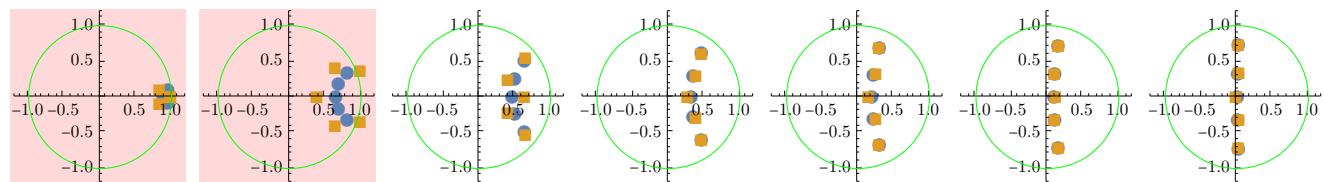
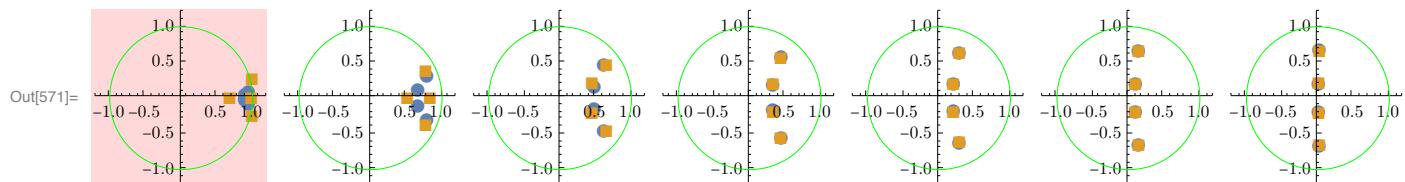
18.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

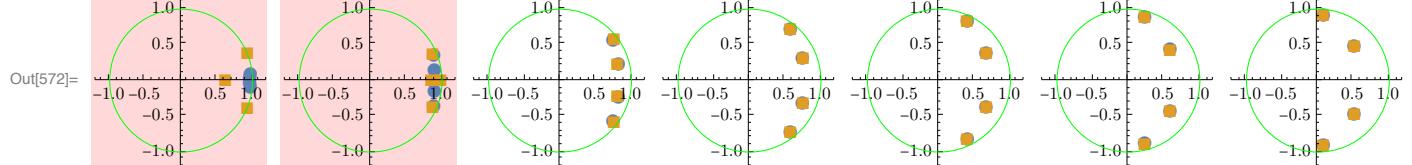
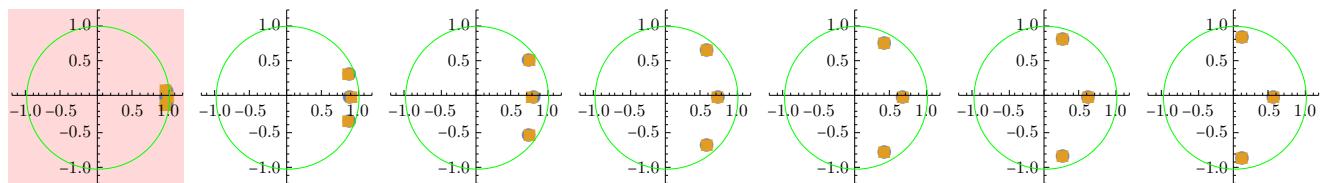
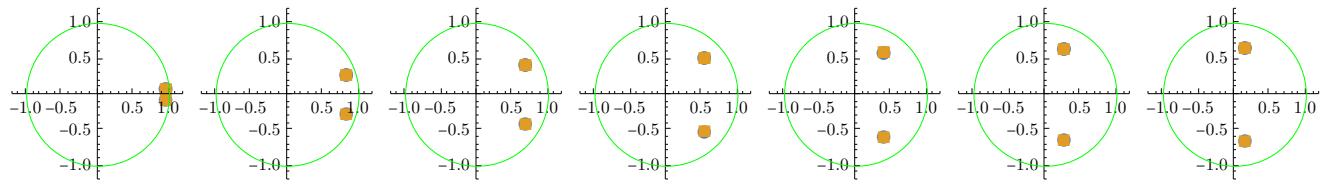
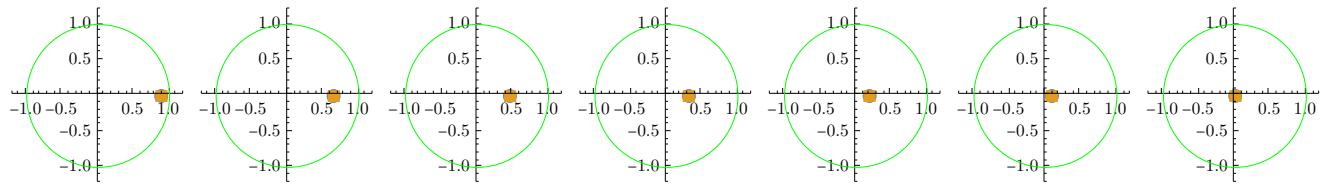
```
In[571]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

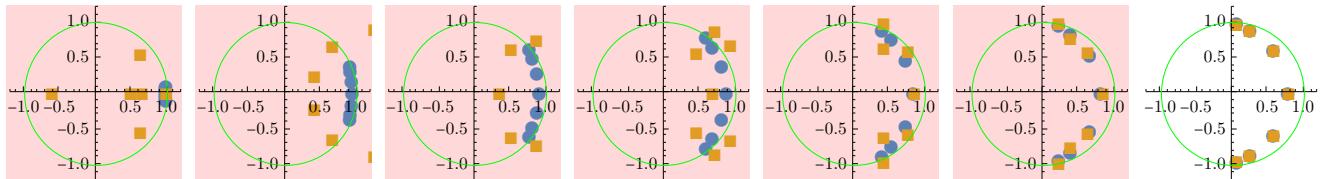
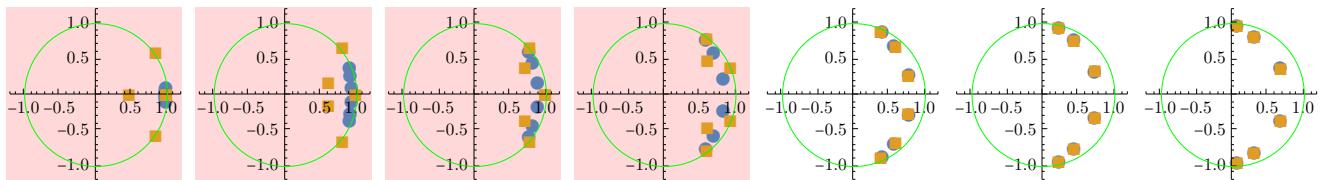
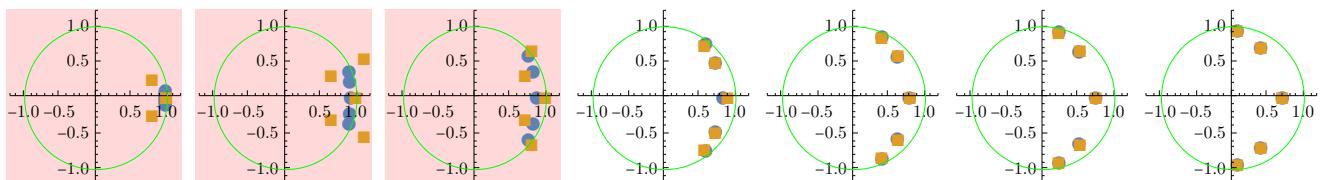




★ Chebyshev 1

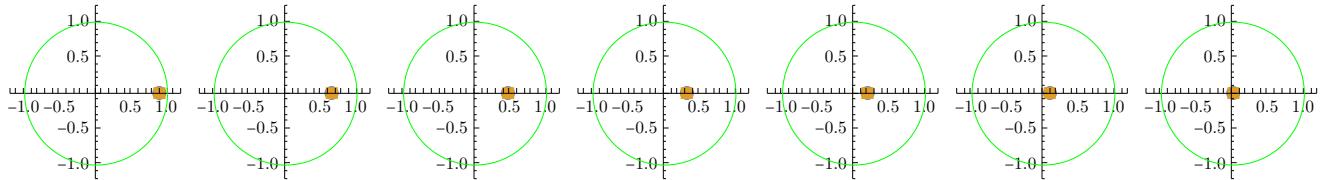
```
In[572]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

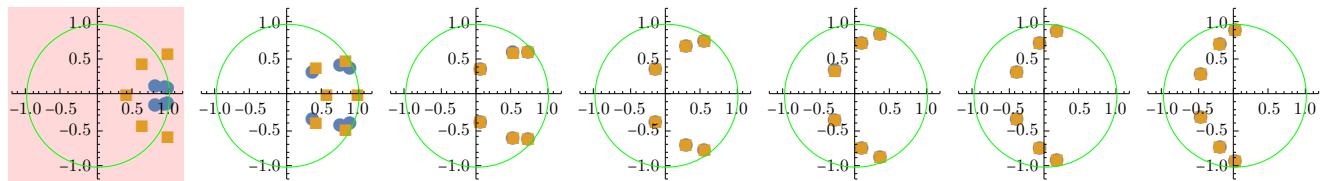
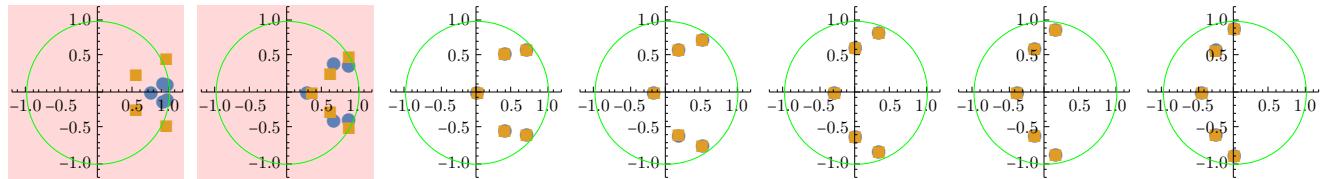
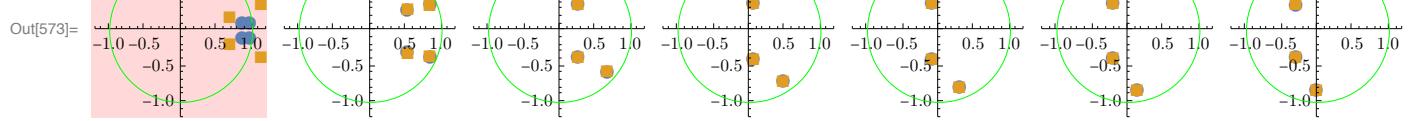
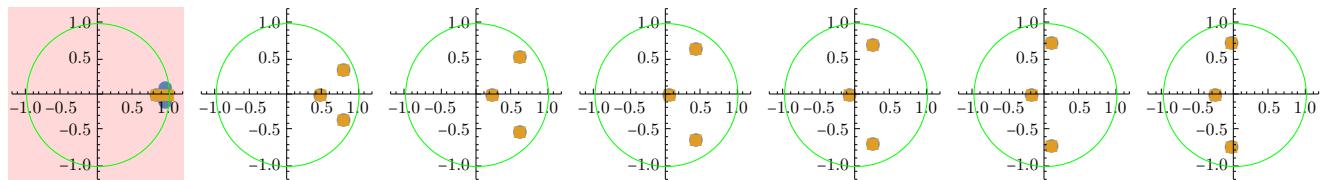
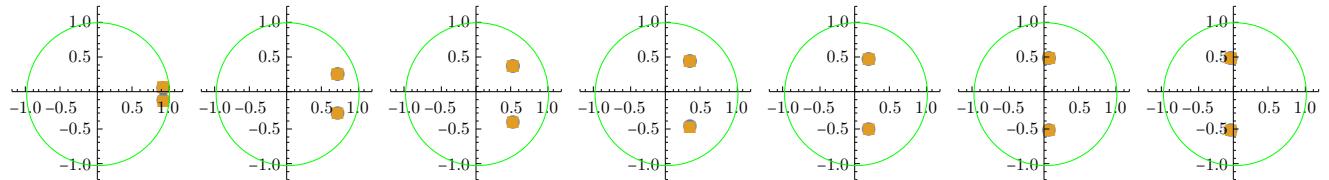


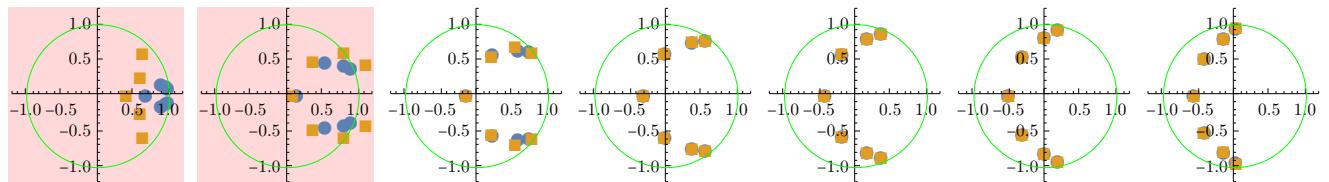


★ Chebyshev 2

```
In[573]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

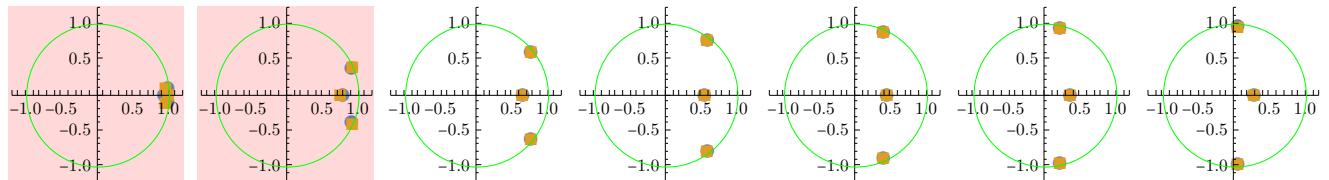
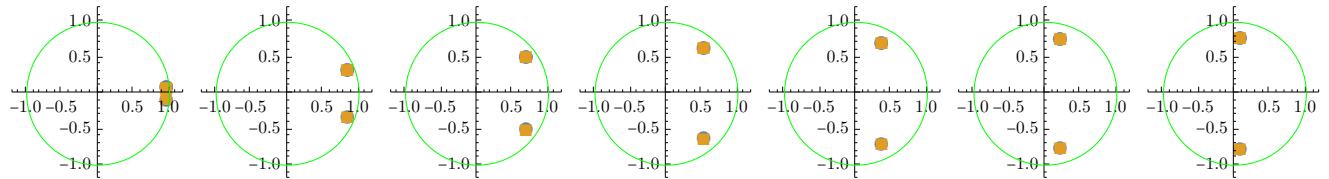
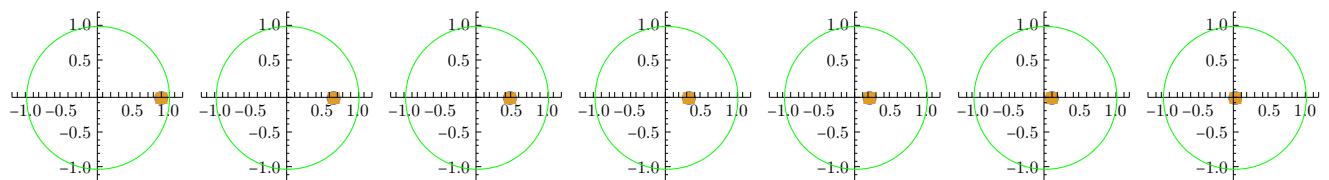


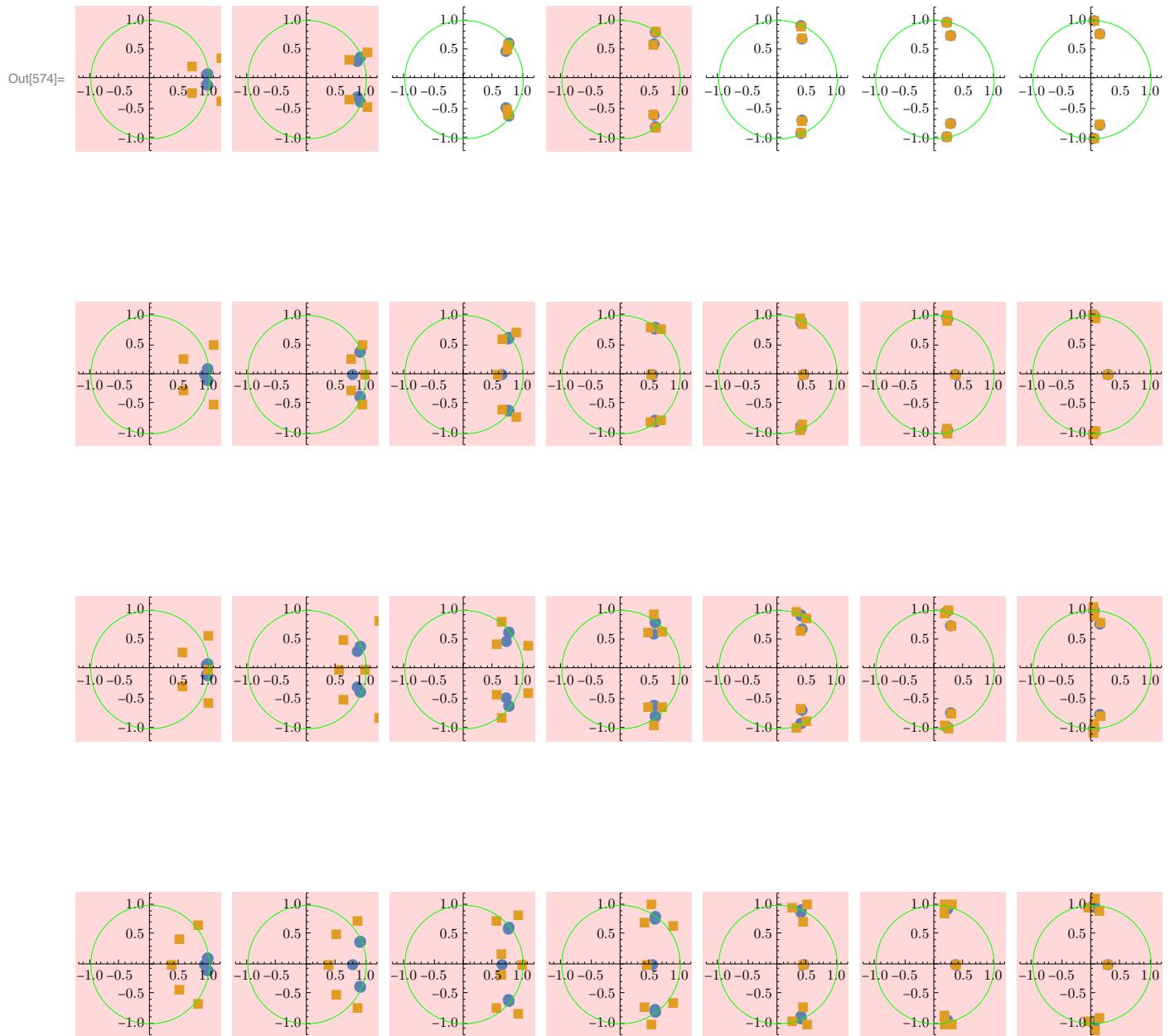




★ Eliptyczne

```
In[574]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





19 Badanie stabilności filtrów cyfrowych (10 bit)

In[575]:= **bity = 10;**

19.1 Dyskretyzacja na poziomie współczynników

19.1.1 Dyskretyzacja

In[576]:= **DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];**

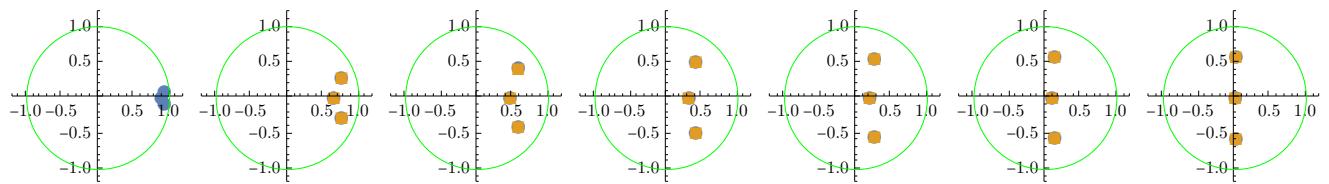
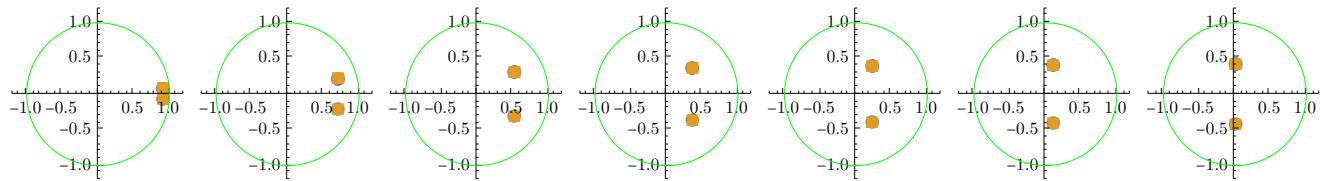
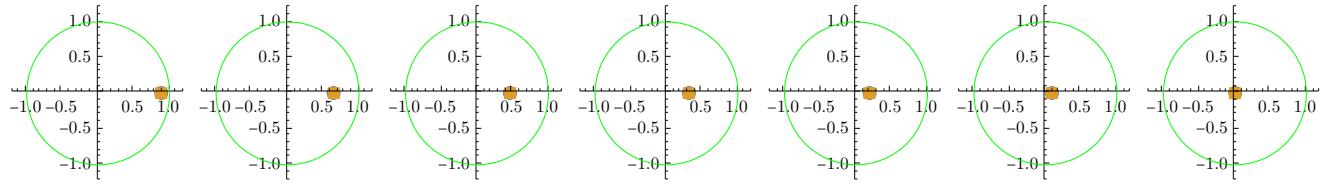
```
In[577]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[578]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[579]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

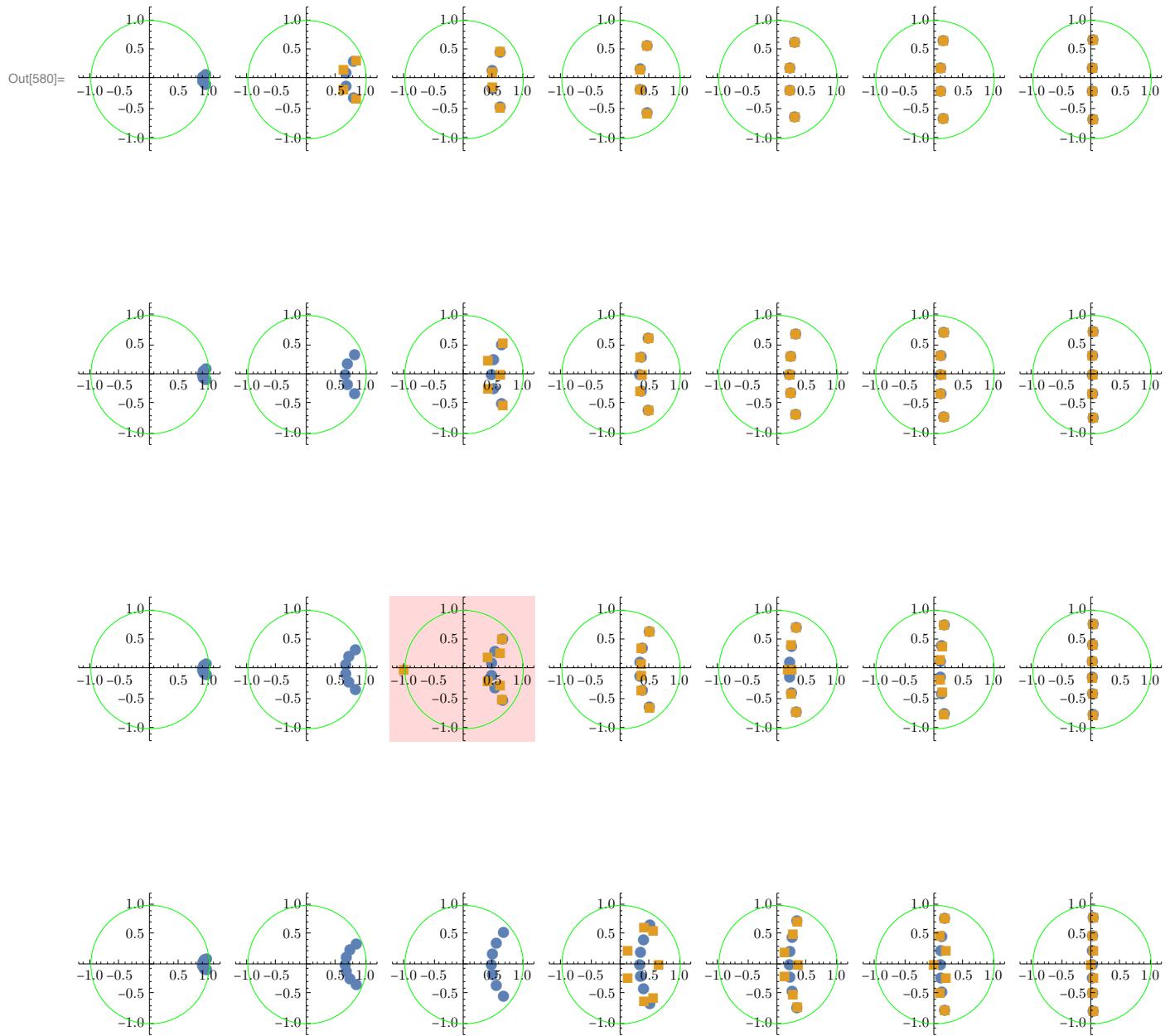
19.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

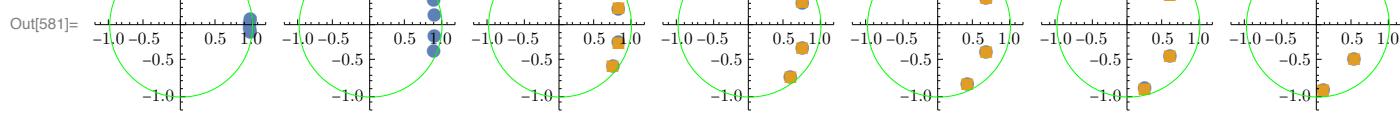
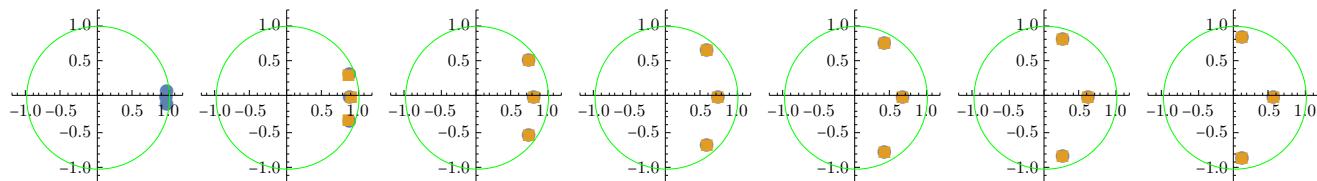
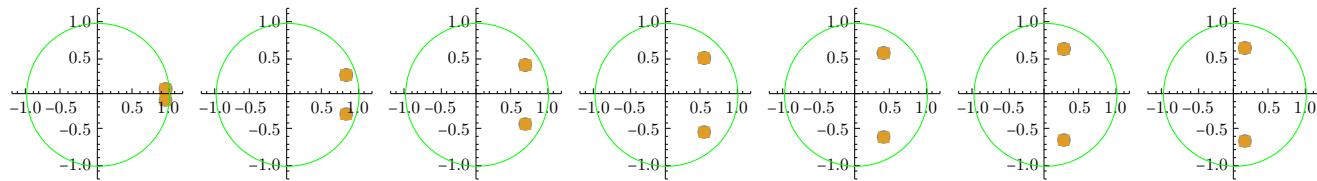
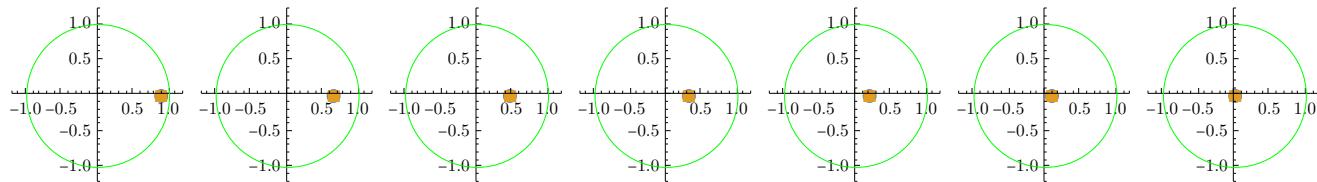
```
In[580]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

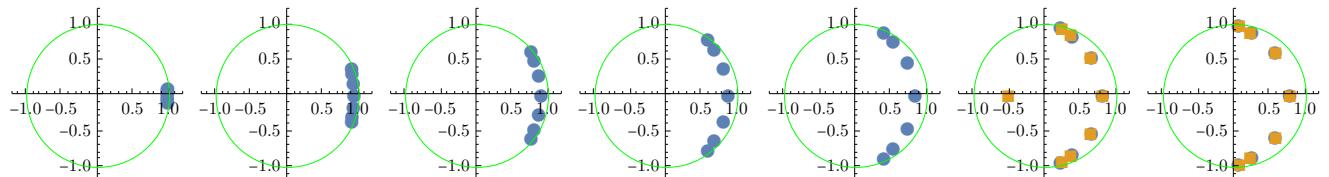
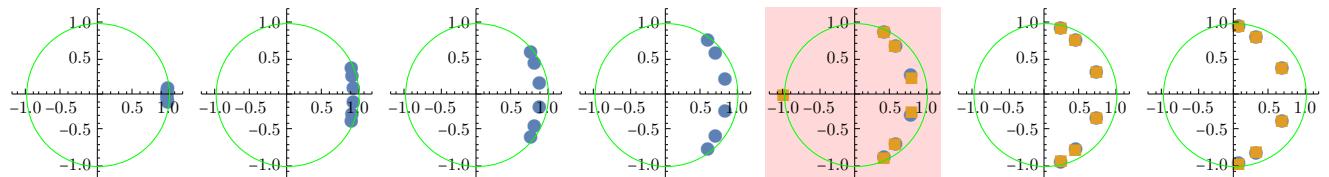
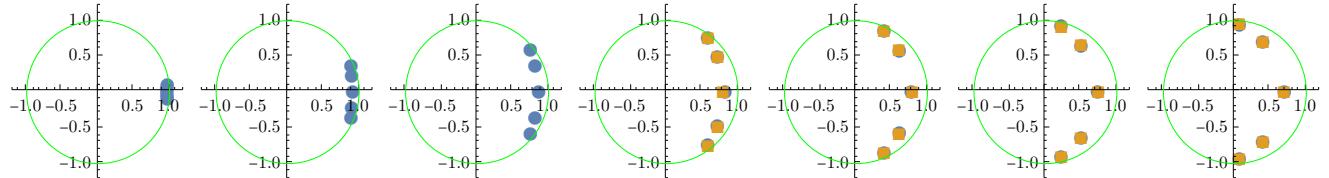




★ Chebyshev 1

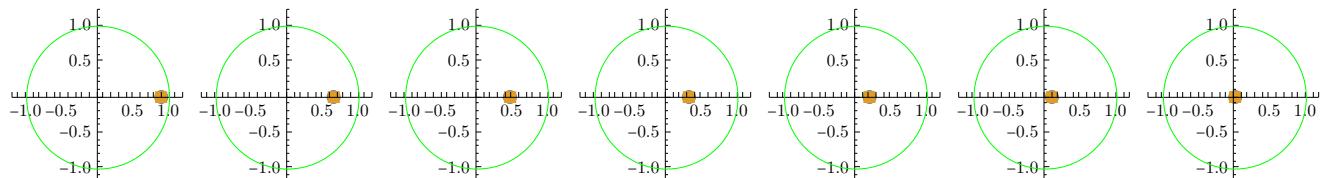
```
In[581]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

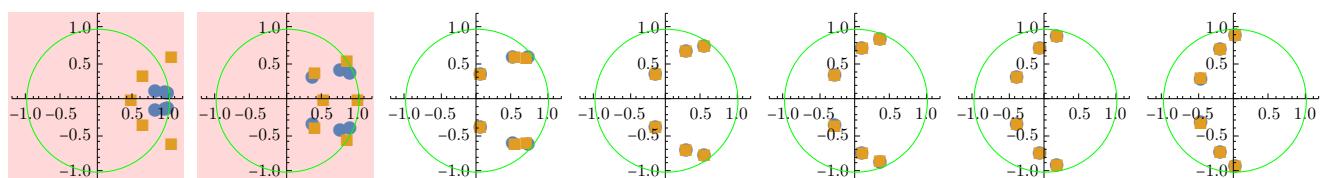
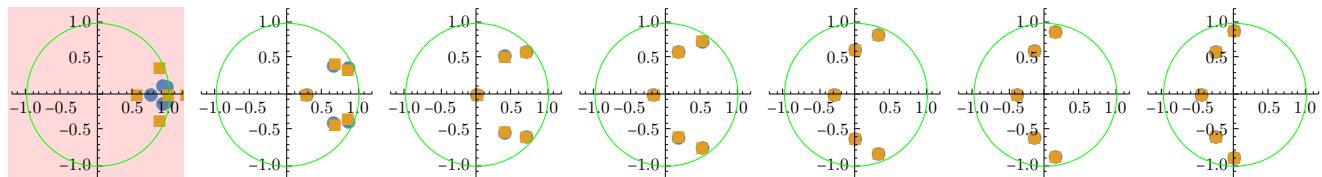
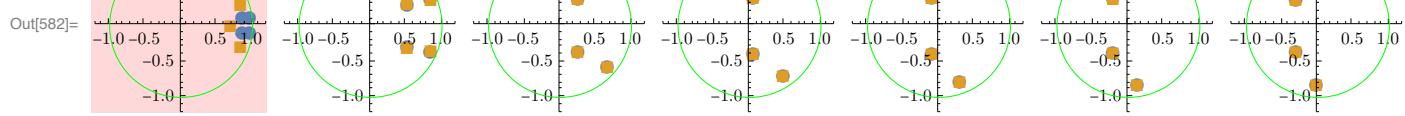
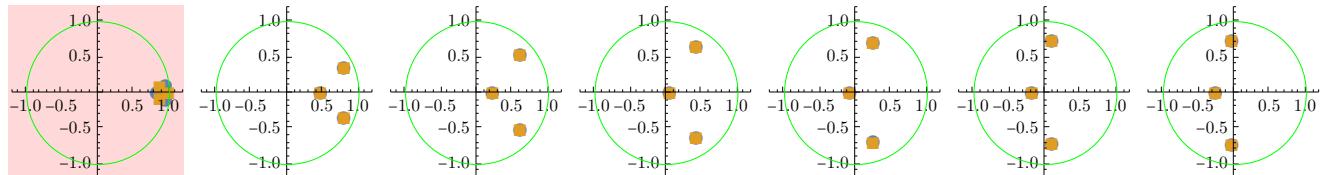
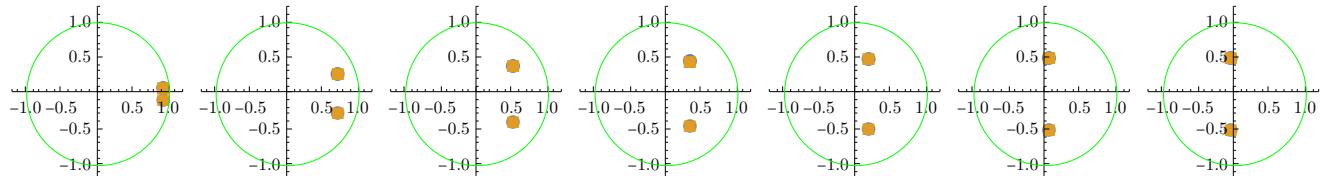


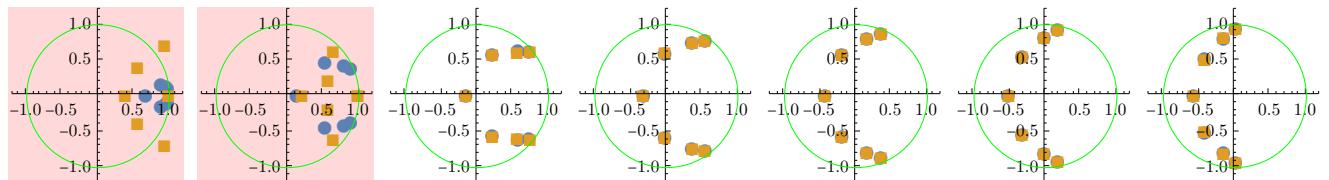


★ Chebyshev 2

```
In[582]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

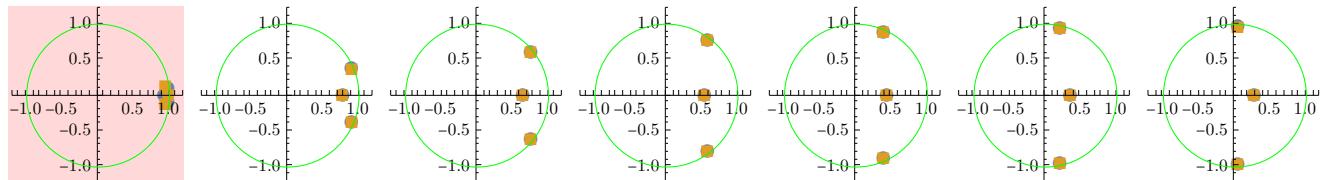
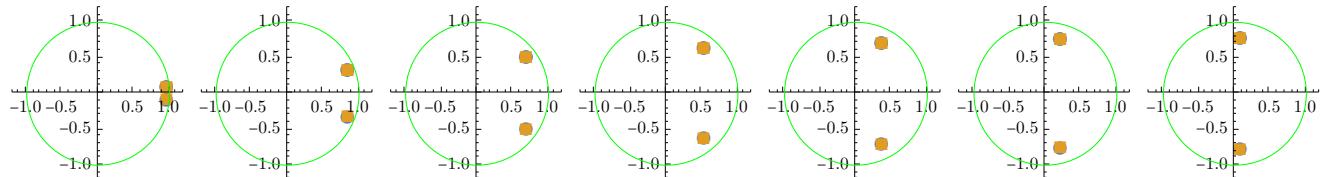
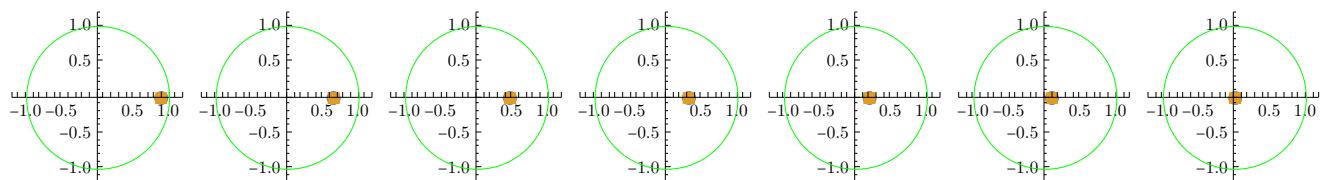


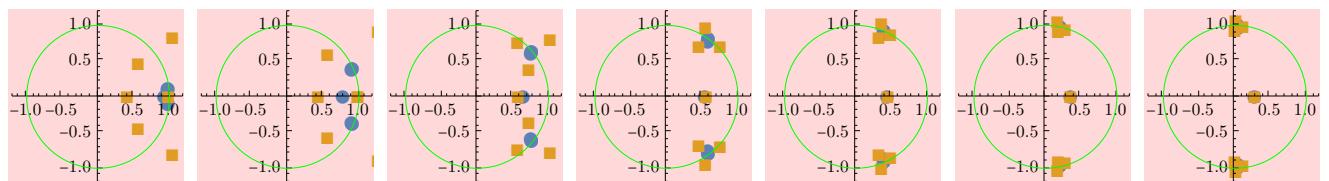
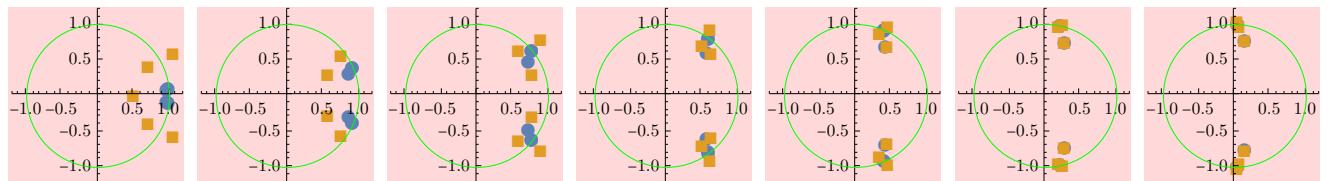
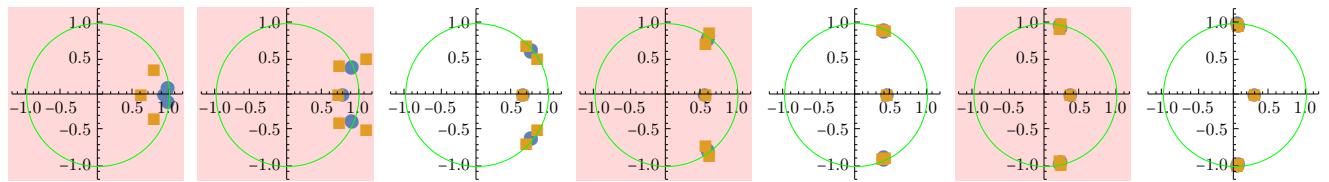
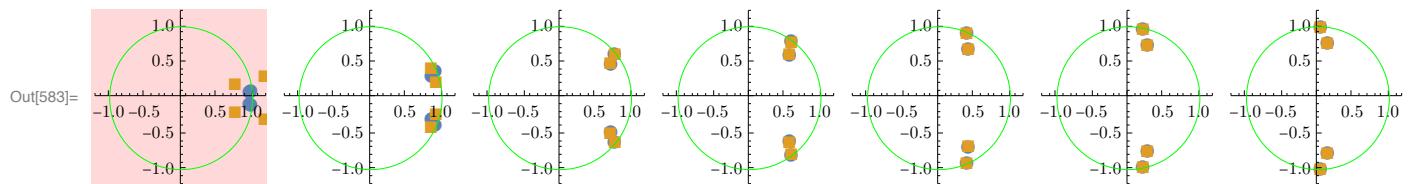




★ Eliptyczne

```
In[583]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





19.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

19.2.1 Dyskretyzacja

```
In[584]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[585]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[586]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

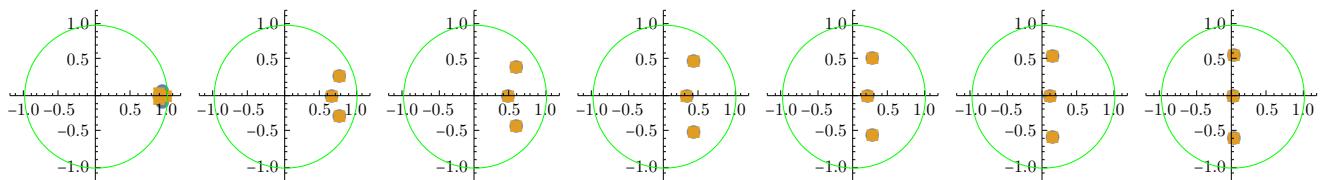
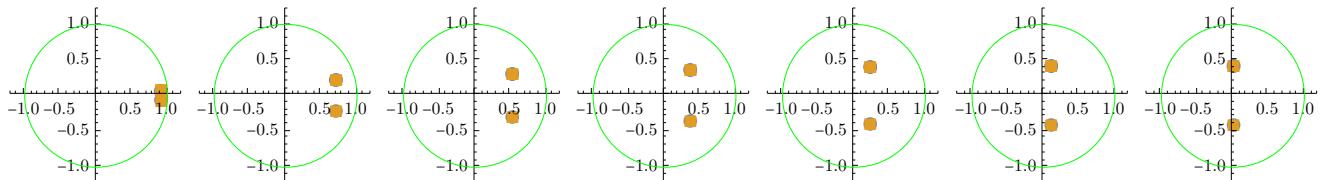
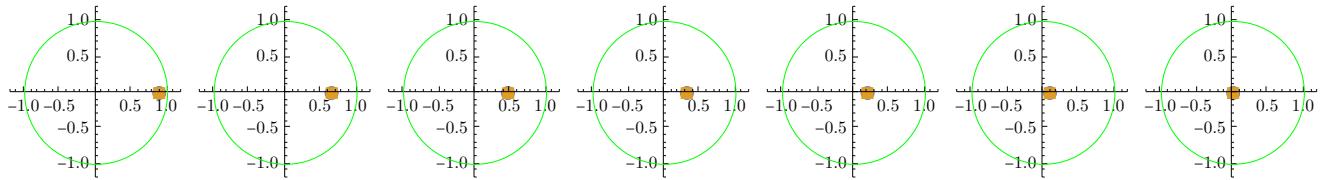
```
In[587]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

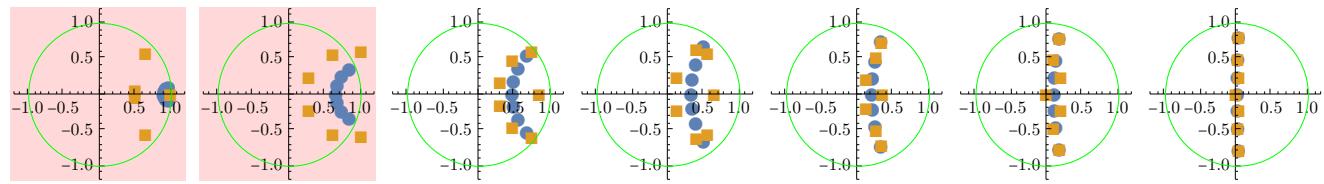
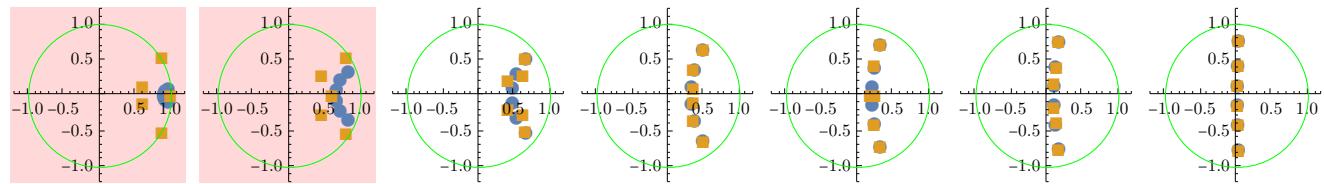
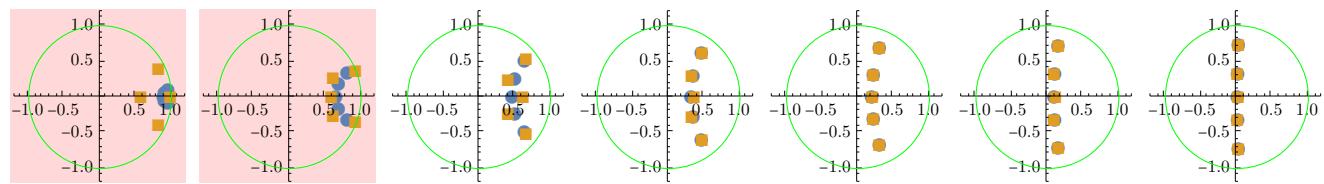
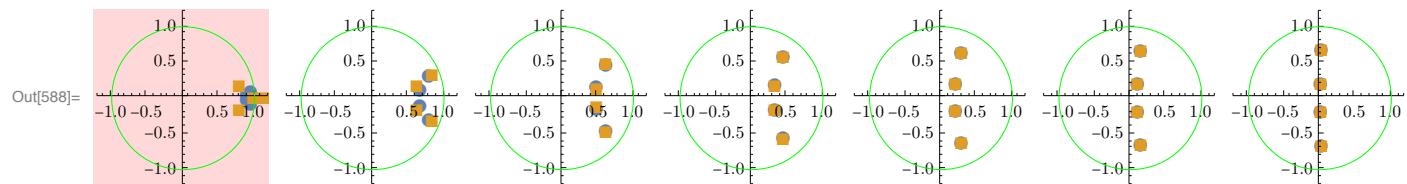
19.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

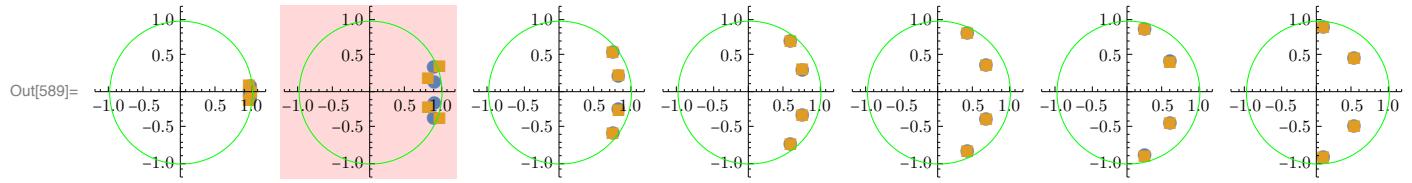
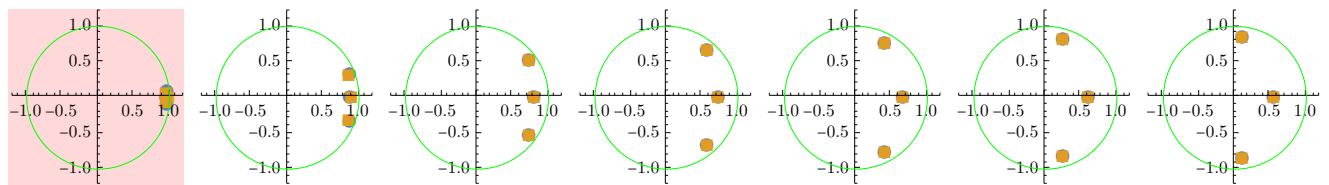
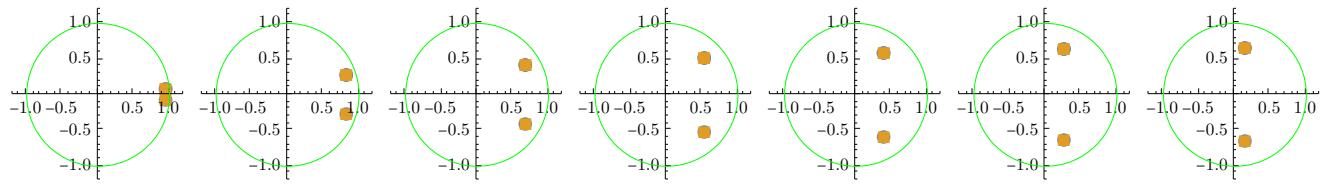
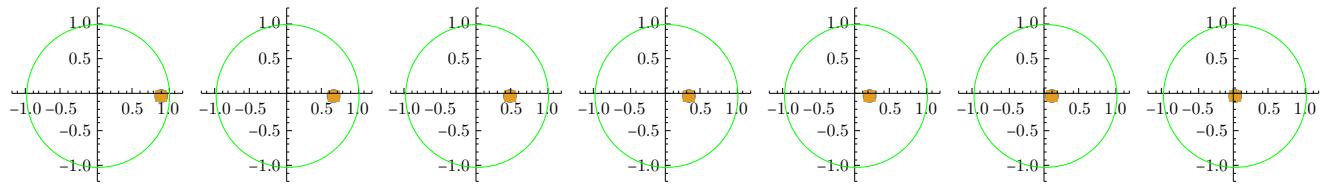
```
In[588]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

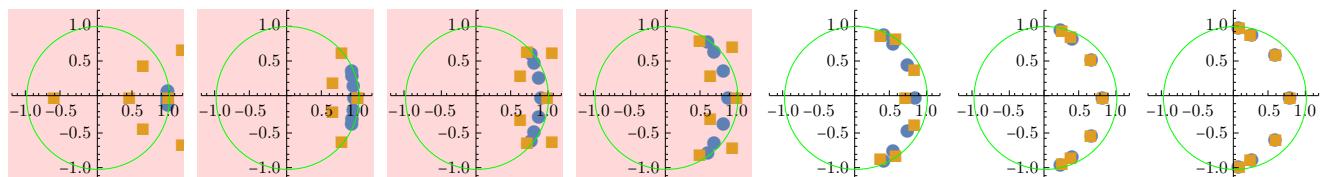
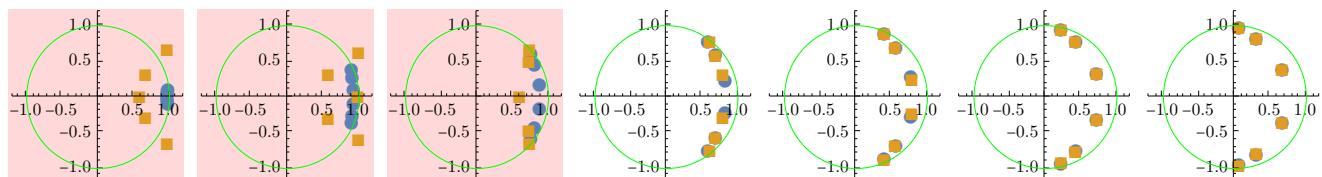
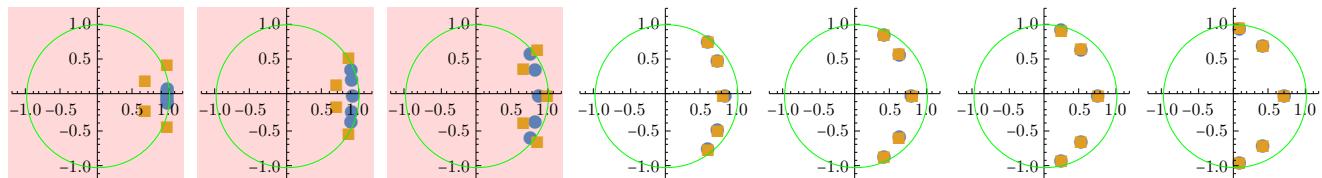




★ Chebyshev 1

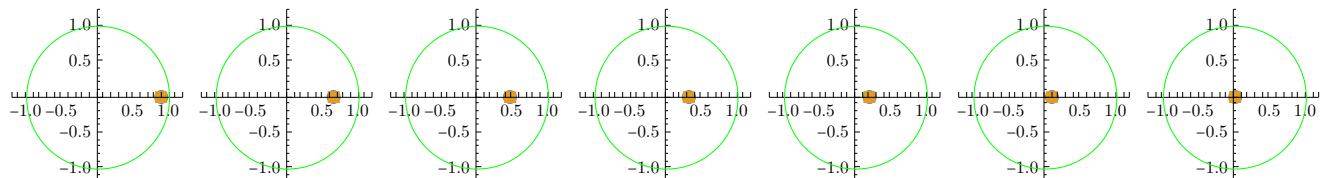
```
In[589]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

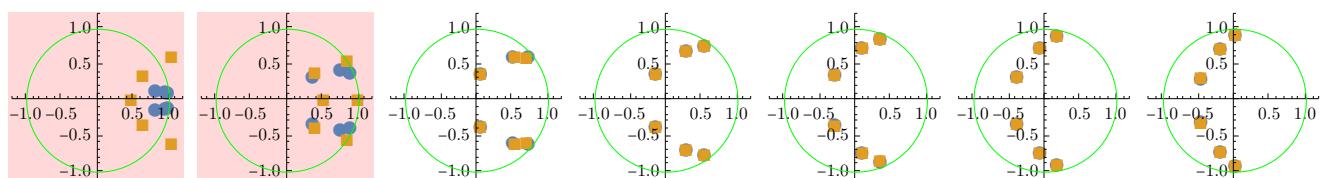
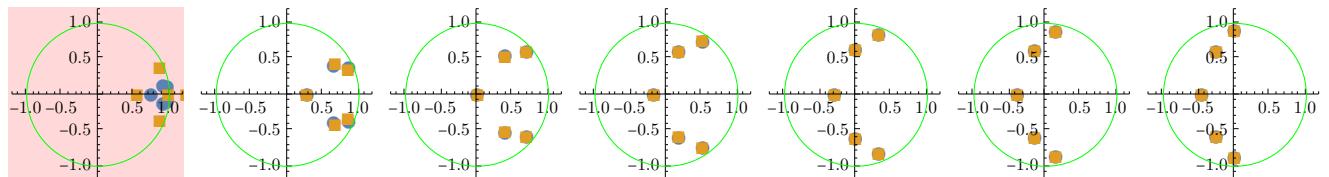
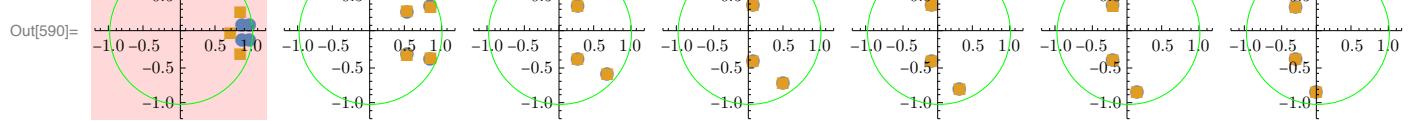
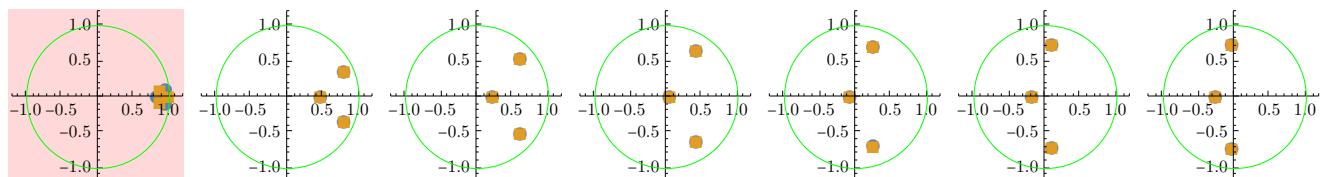
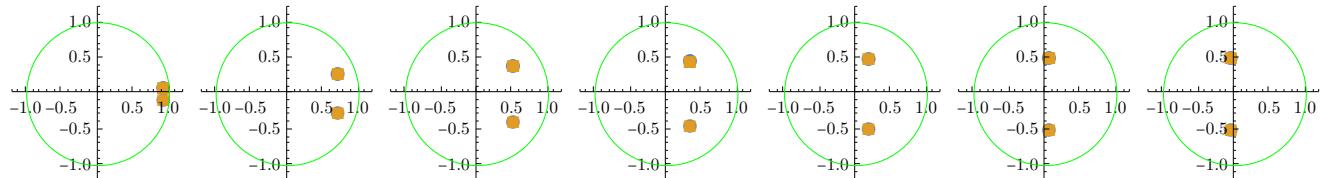


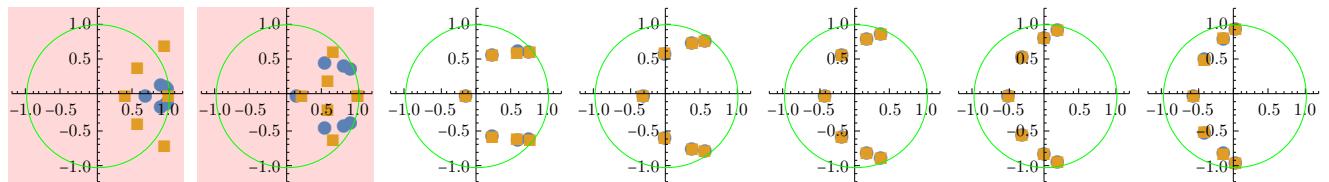


★ Chebyshev 2

```
In[590]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

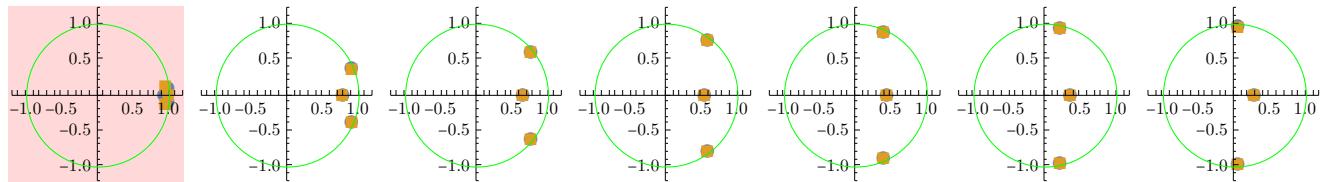
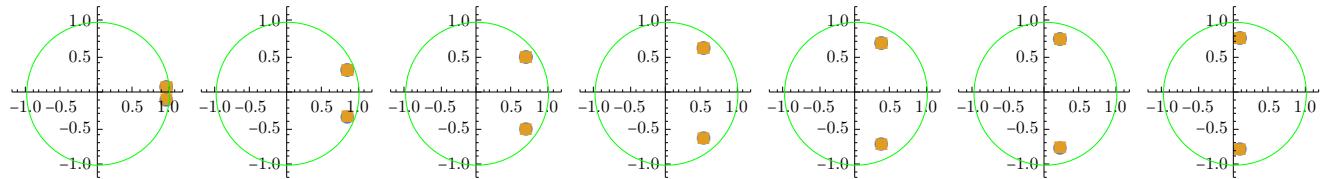
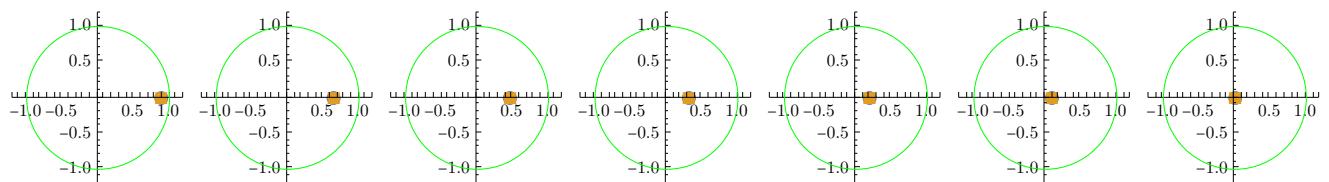


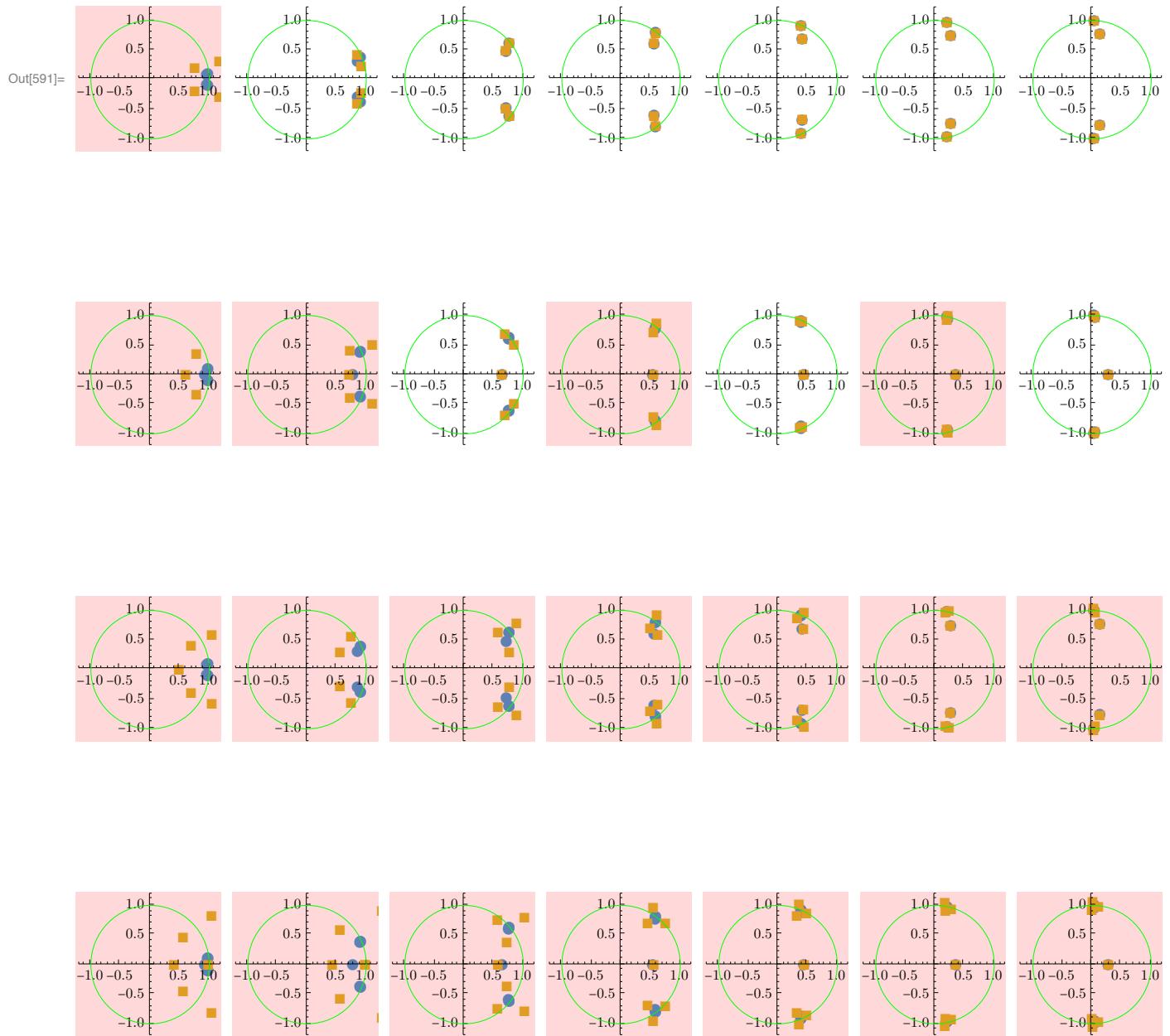




★ Eliptyczne

```
In[591]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





20 Badanie stabilności filtrów cyfrowych (11 bit)

```
In[592]:= bity = 11;
```

20.1 Dyskretyzacja na poziomie współczynników

20.1.1 Dyskretyzacja

```
In[593]:= DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];
```

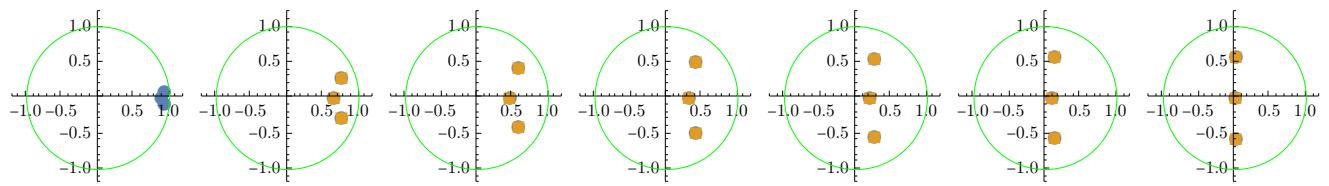
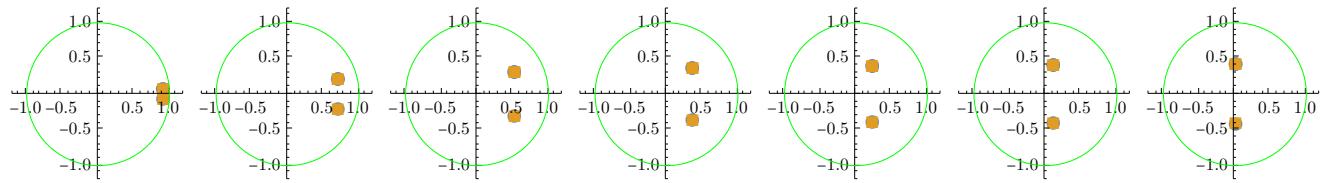
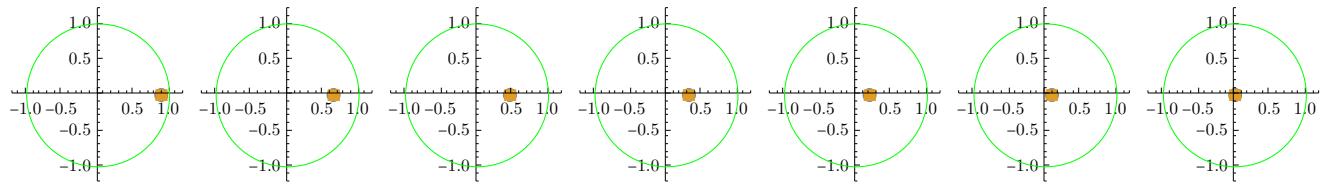
```
In[594]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[595]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[596]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

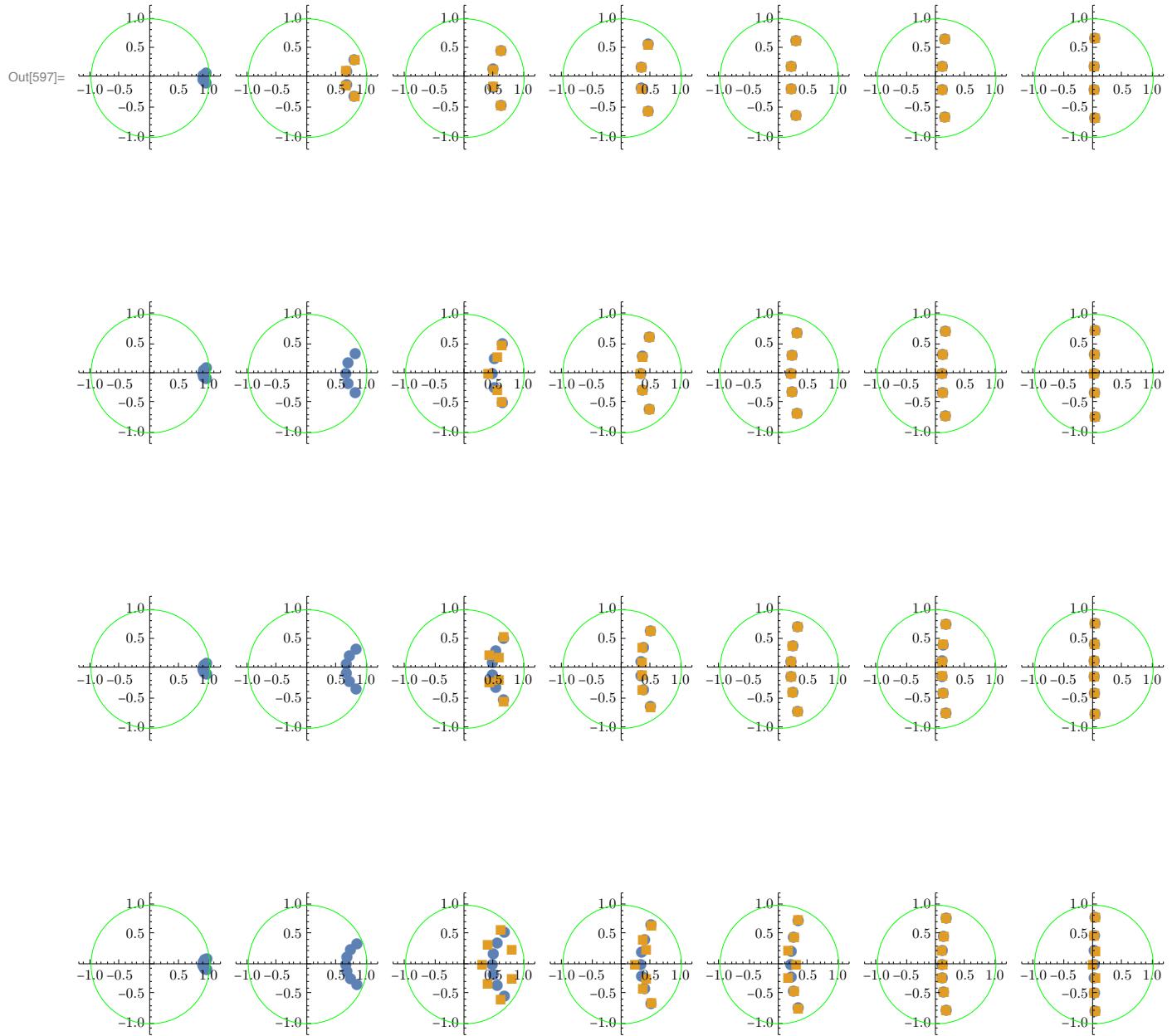
20.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

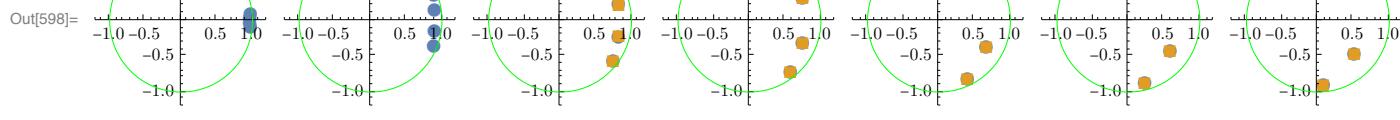
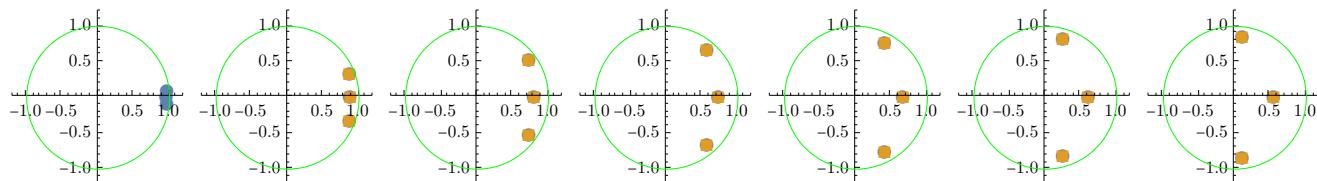
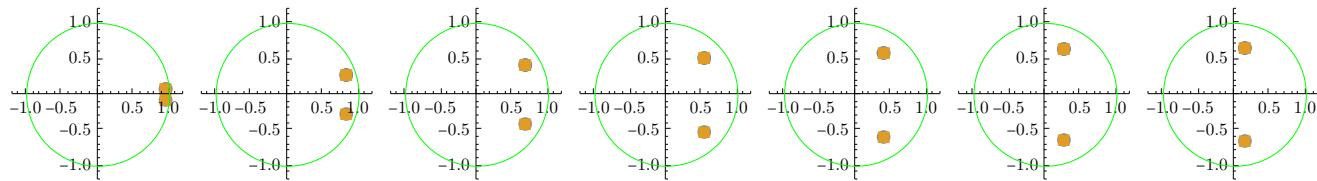
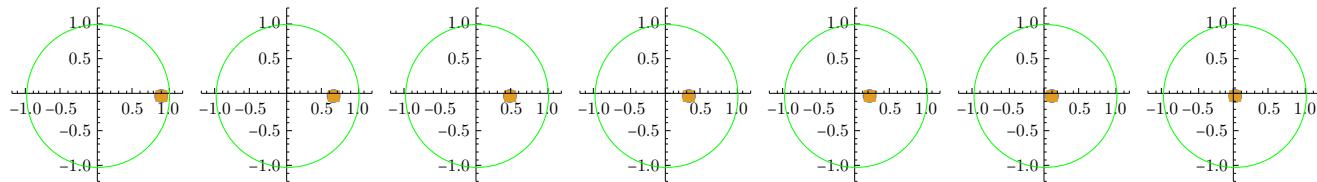
```
In[597]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

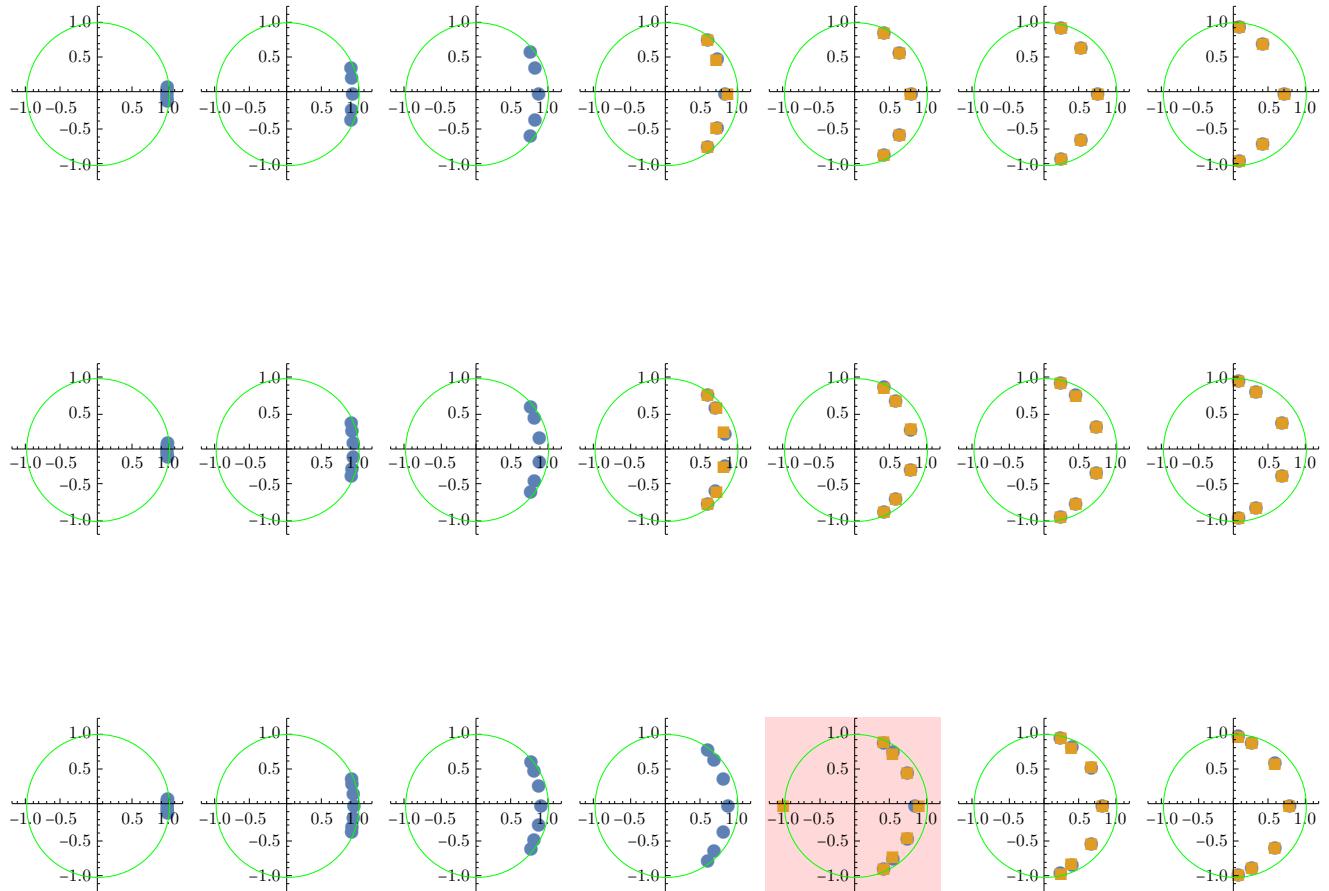




★ Chebyshev 1

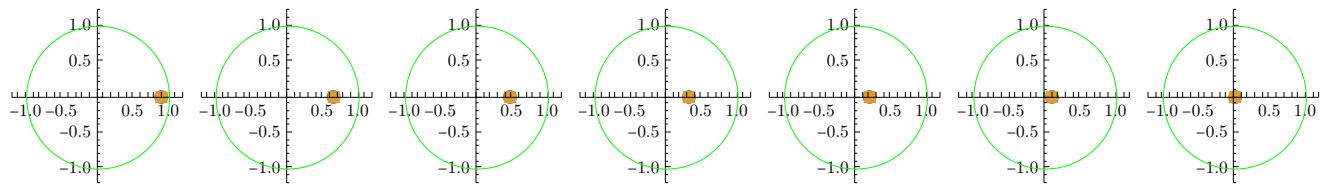
```
In[598]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

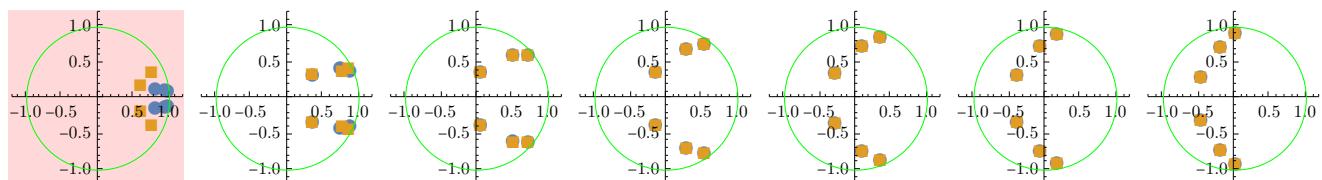
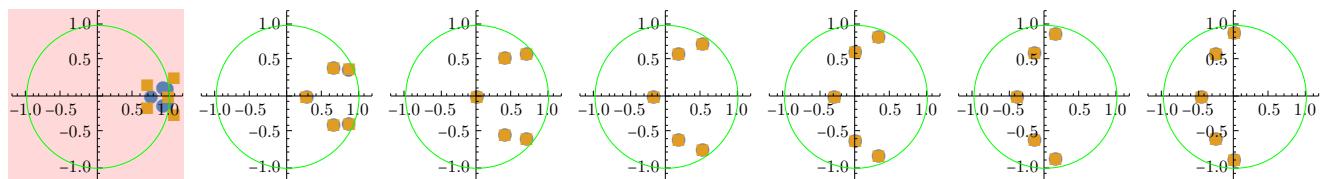
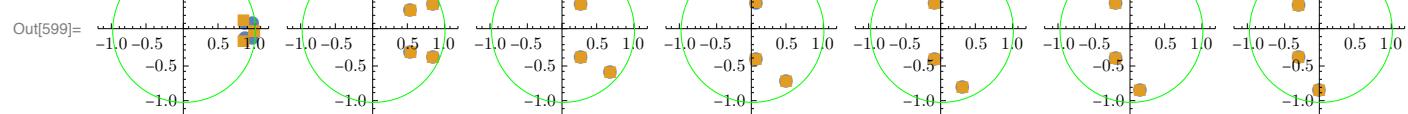
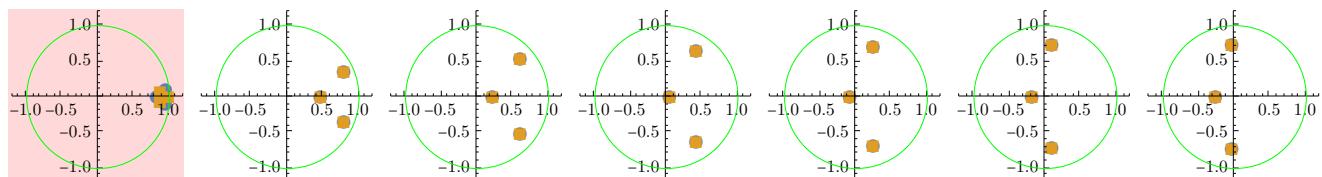
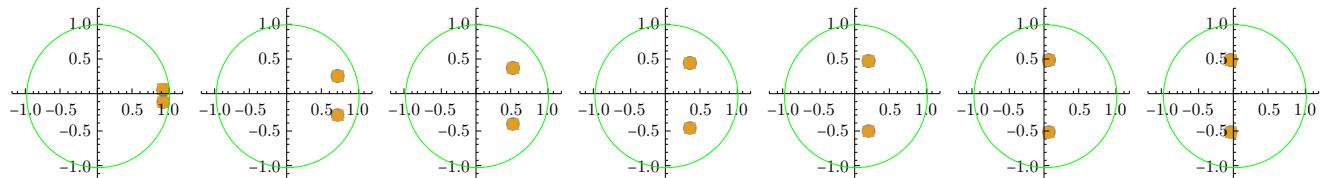


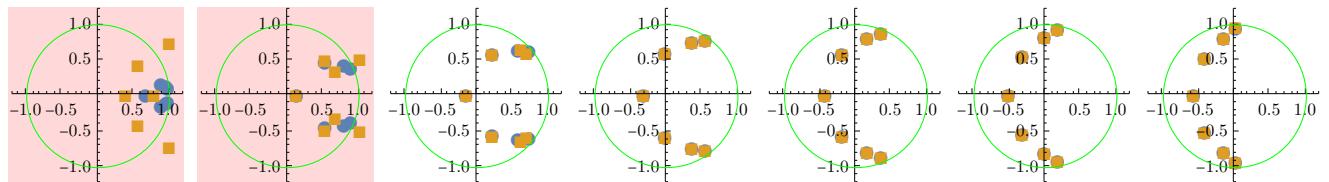


★ Chebyshev 2

```
In[599]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

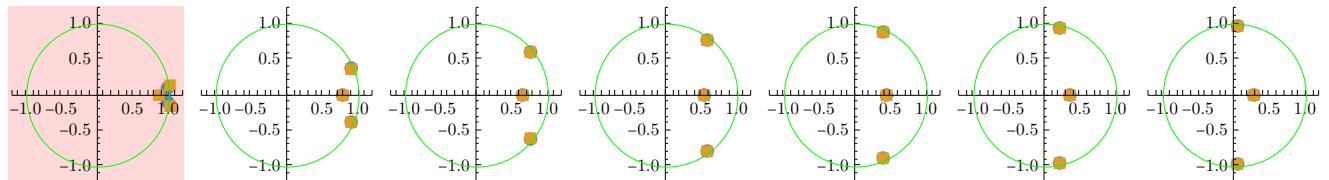
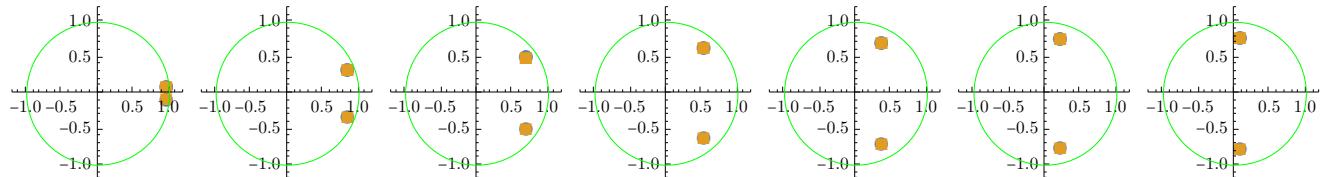
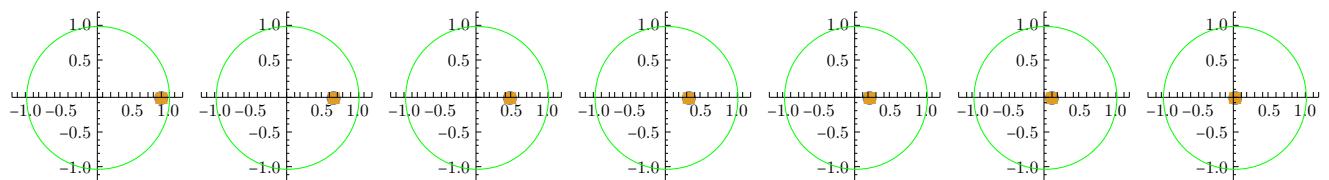


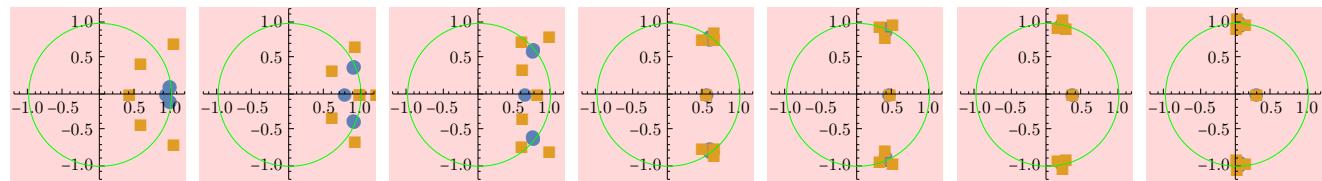
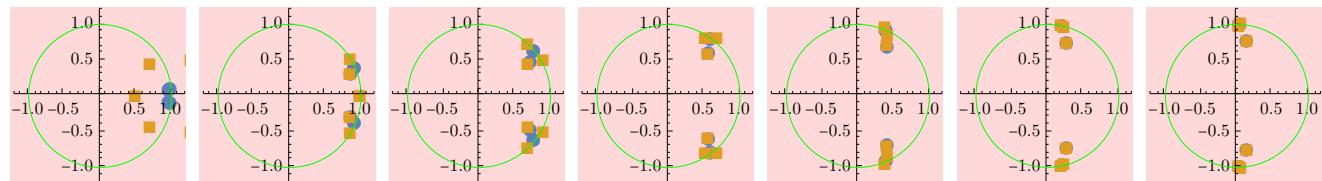
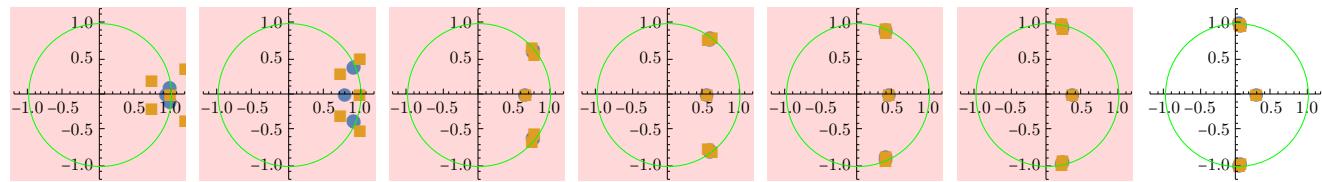
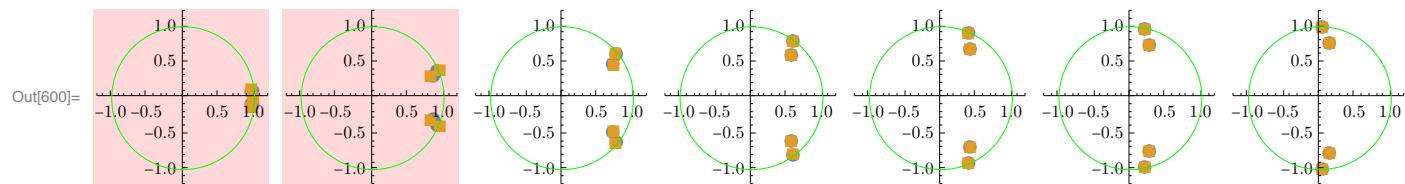




★ Eliptyczne

```
In[600]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





20.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

20.2.1 Dyskretyzacja

```
In[601]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[602]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[603]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

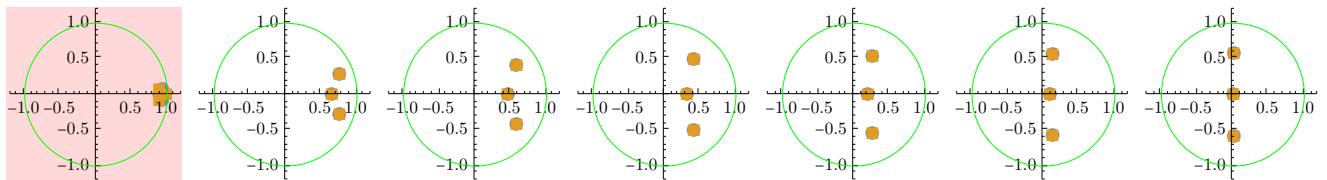
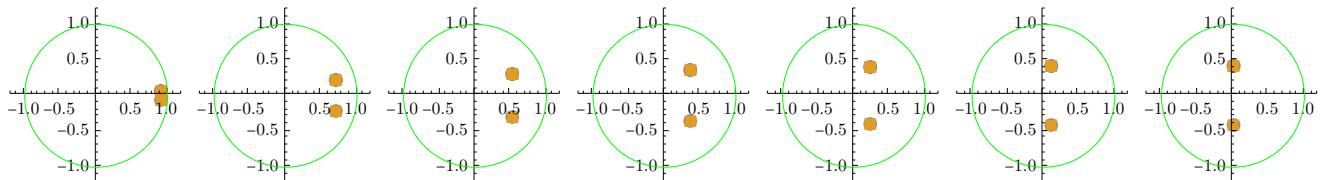
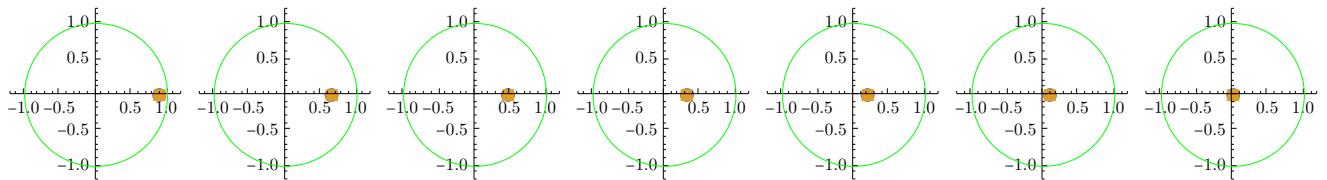
```
In[604]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

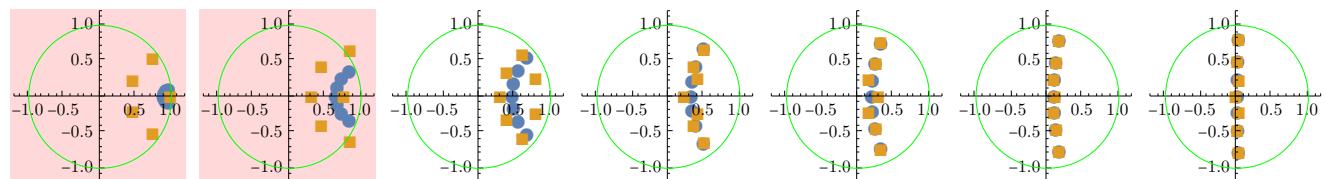
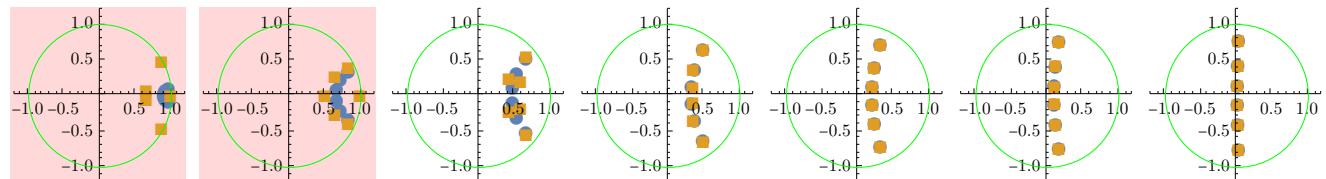
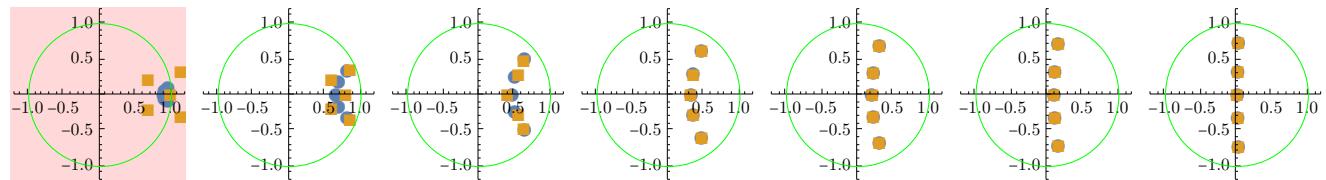
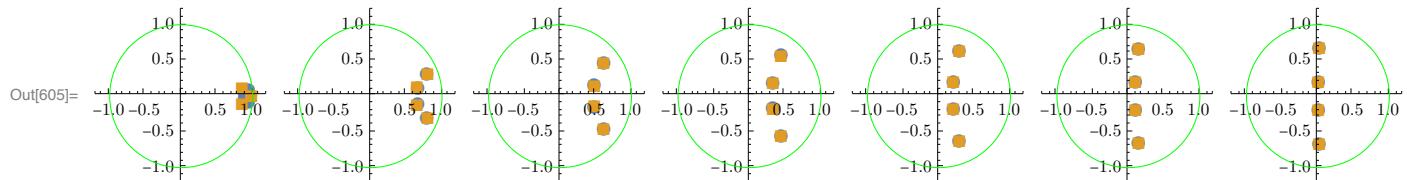
20.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

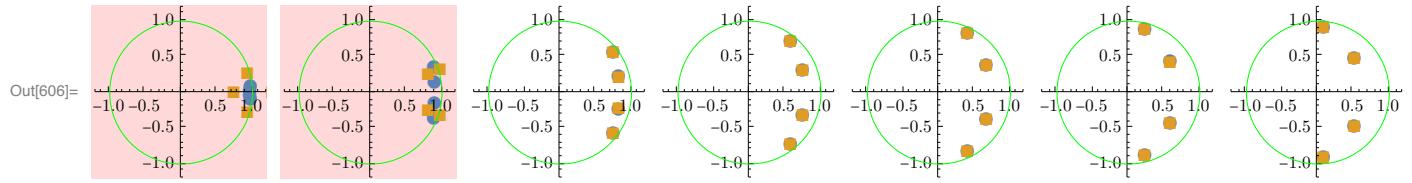
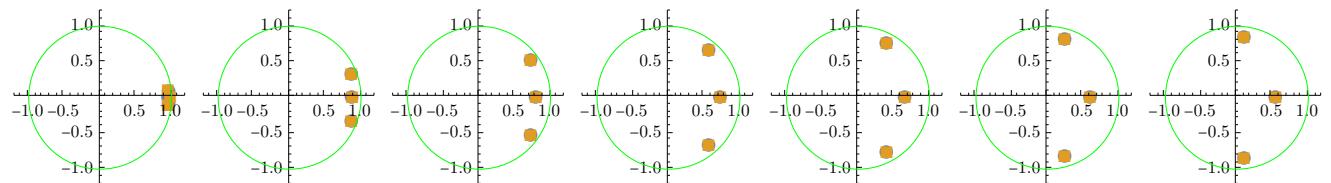
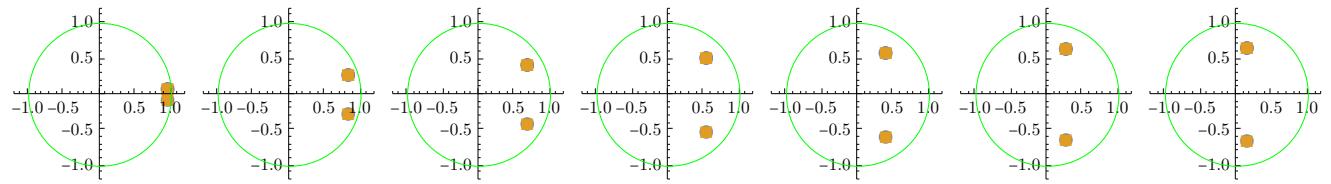
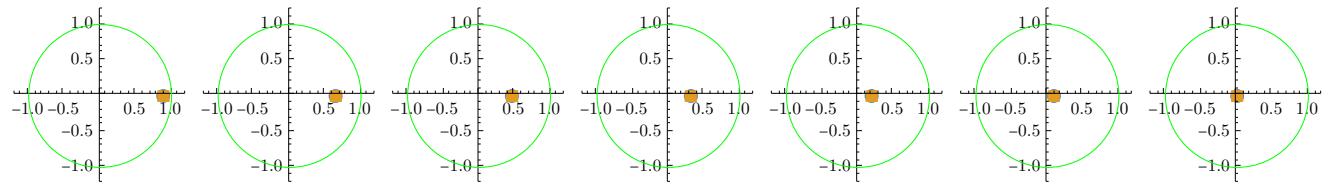
```
In[605]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

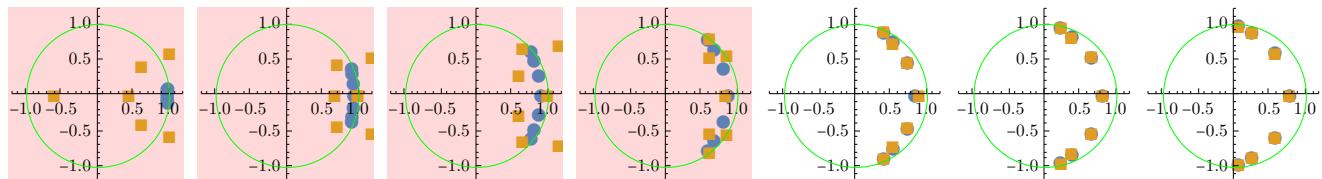
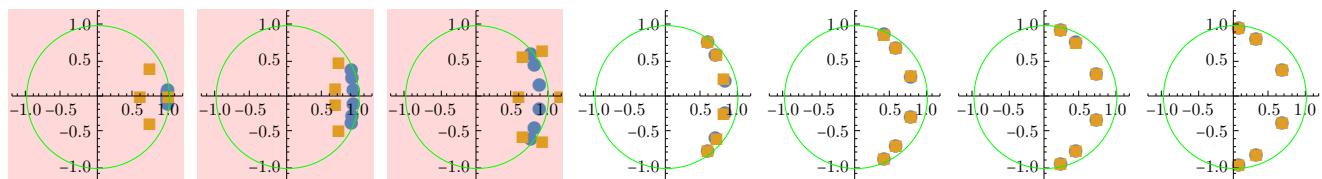
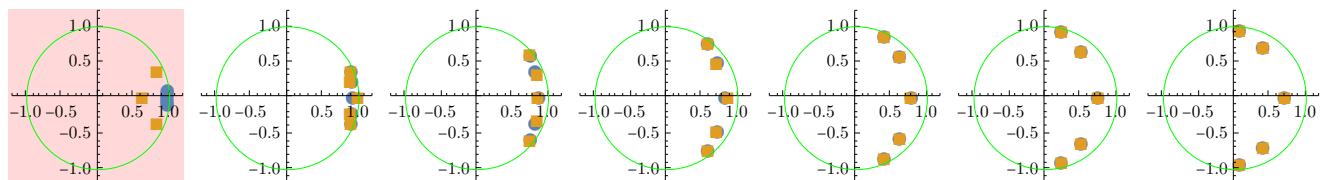




★ Chebyshev 1

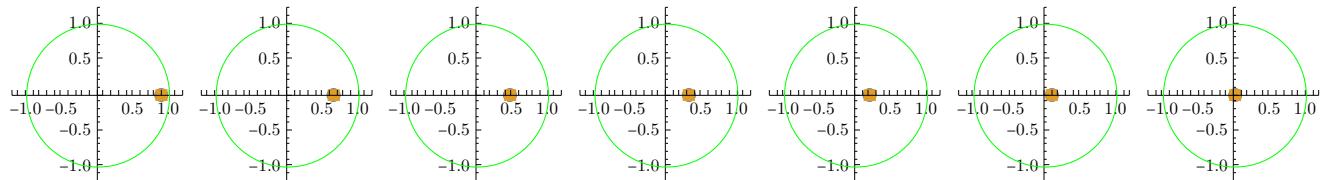
```
In[606]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

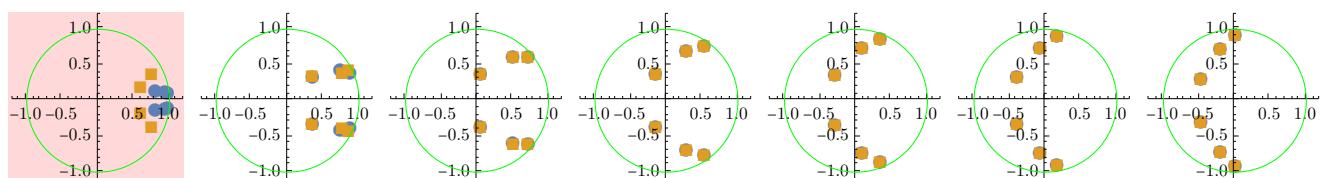
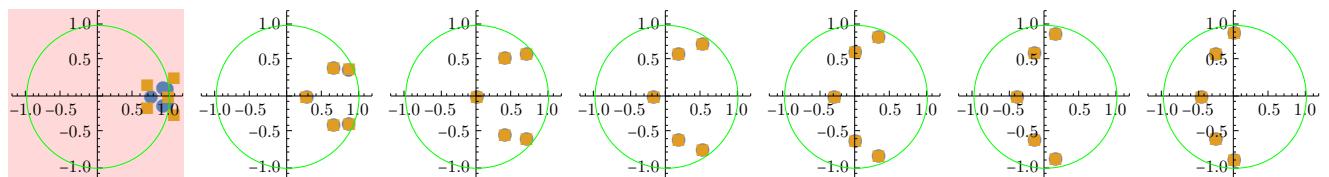
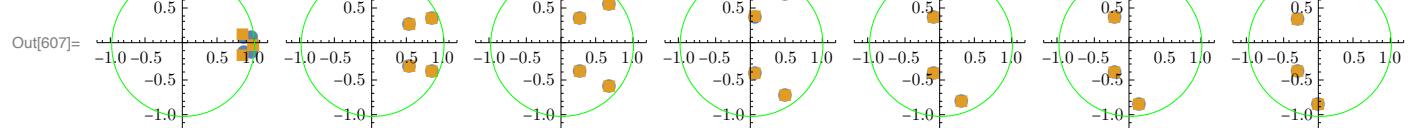
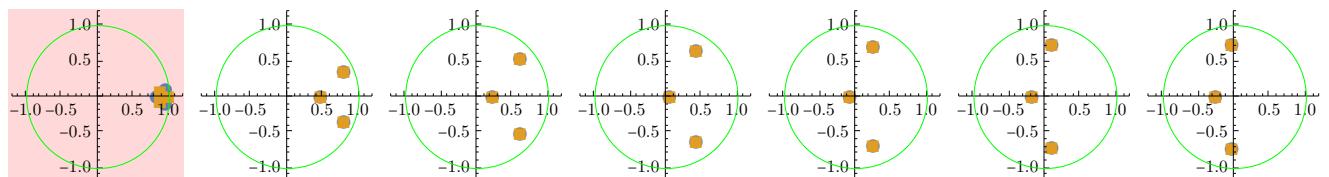
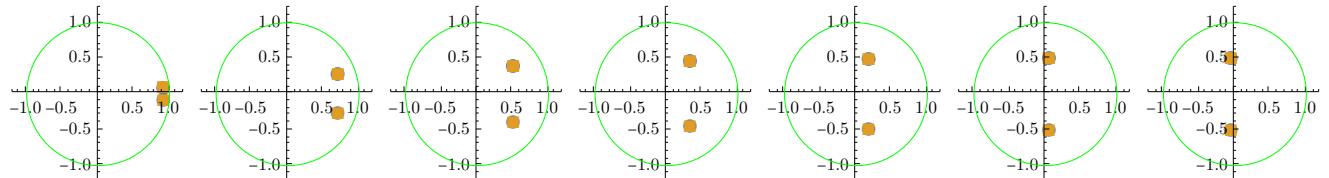


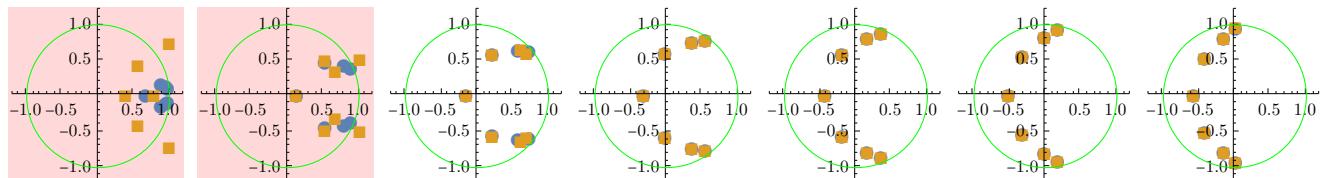


★ Chebyshev 2

```
In[607]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

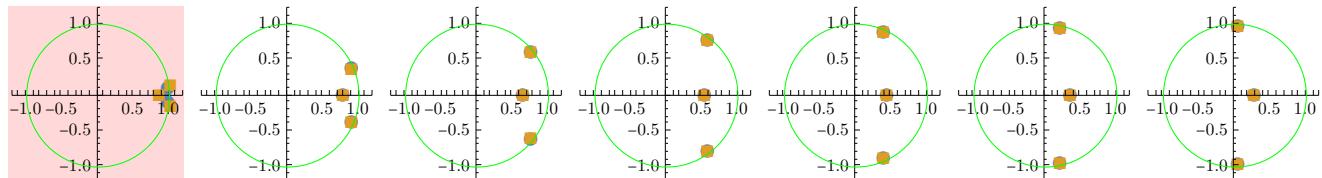
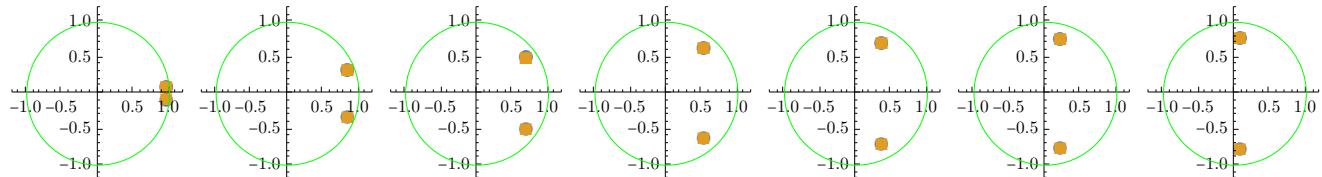
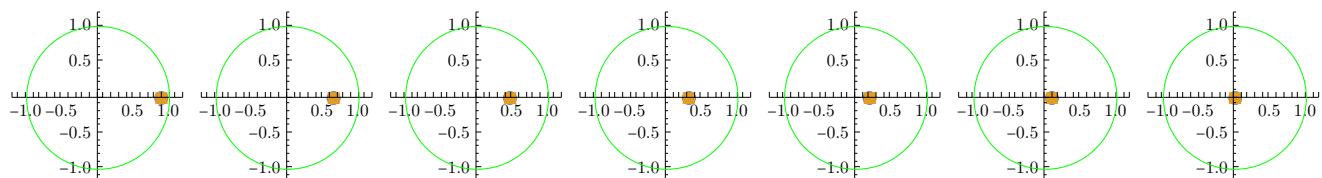






★ Eliptyczne

```
In[608]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





21 Badanie stabilności filtrów cyfrowych (12 bit)

In[609]:= **bity = 12;**

21.1 Dyskretyzacja na poziomie współczynników

21.1.1 Dyskretyzacja

In[610]:= **DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];**

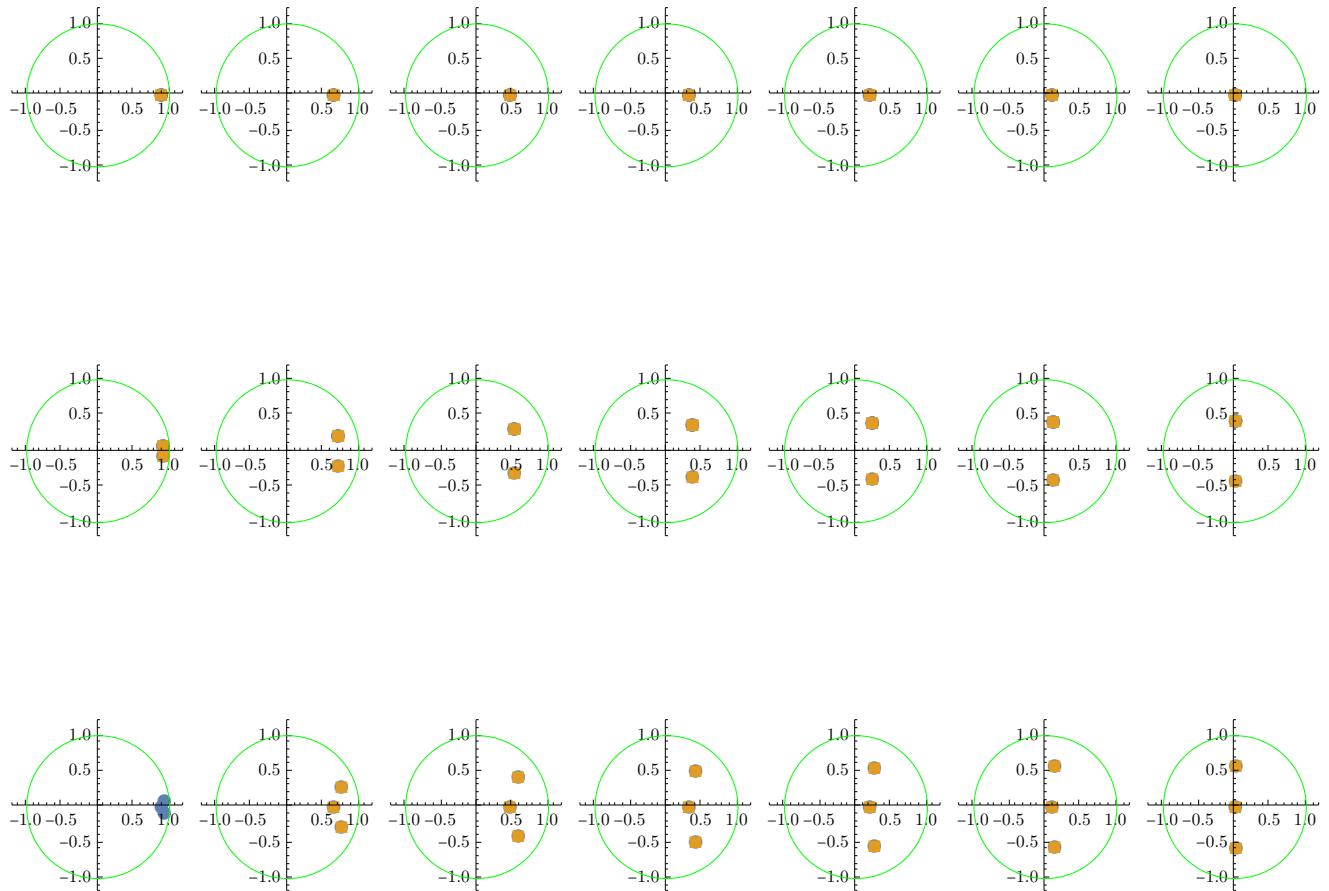
```
In[611]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[612]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[613]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

21.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

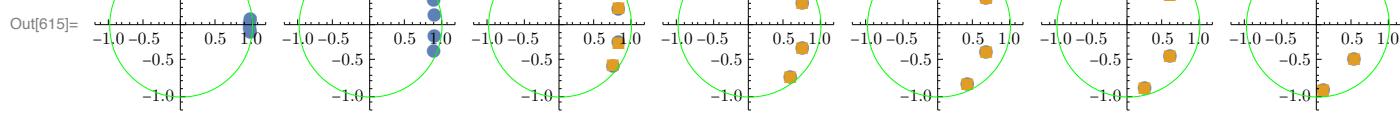
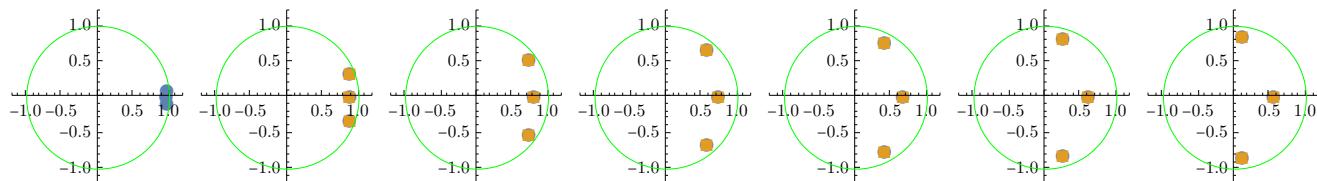
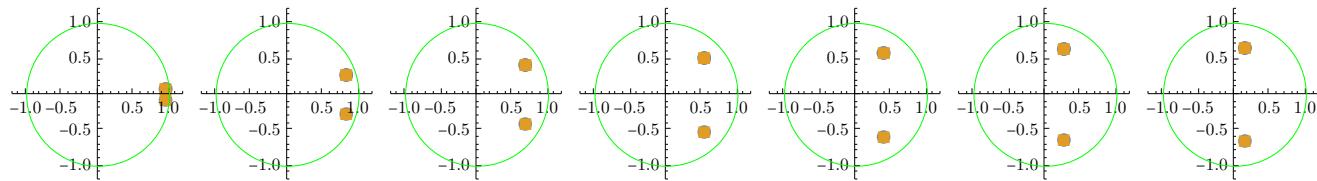
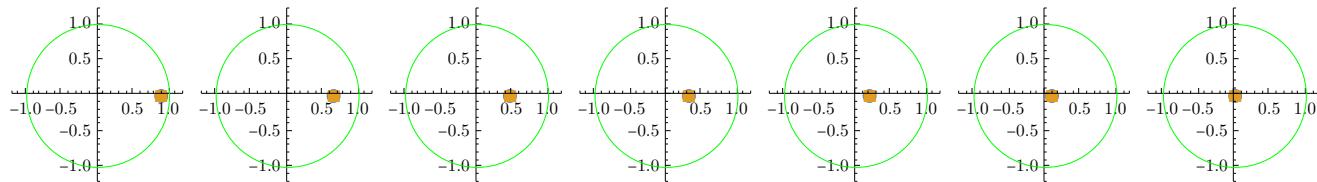
```
In[614]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

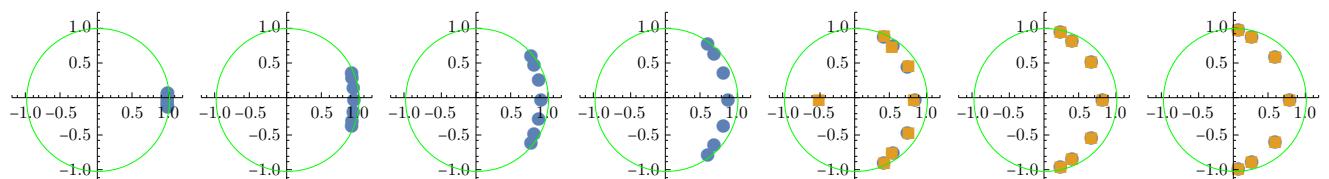
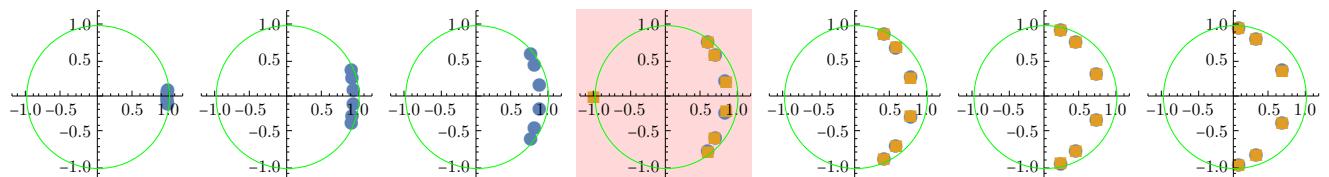
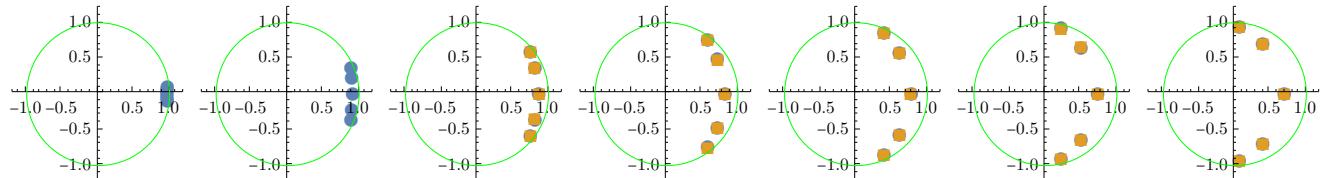




★ Chebyshev 1

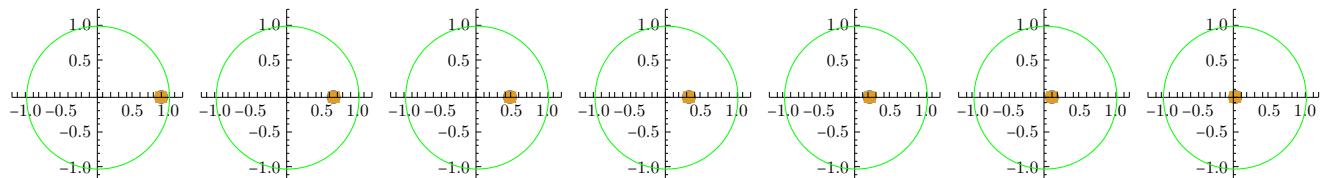
```
In[615]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

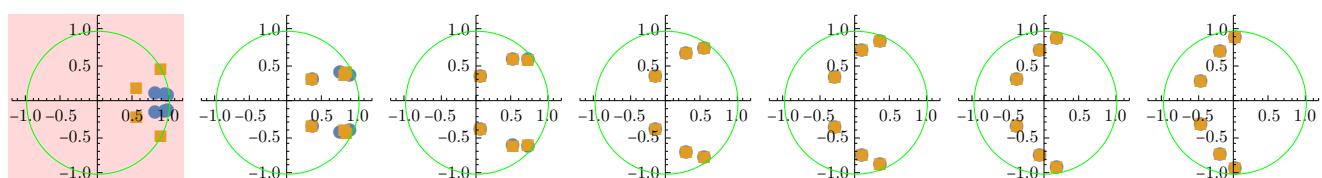
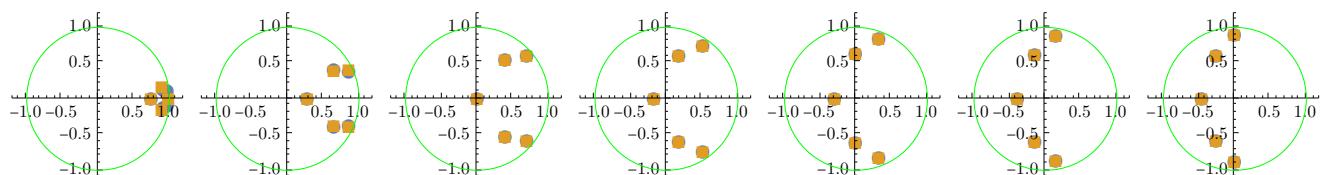
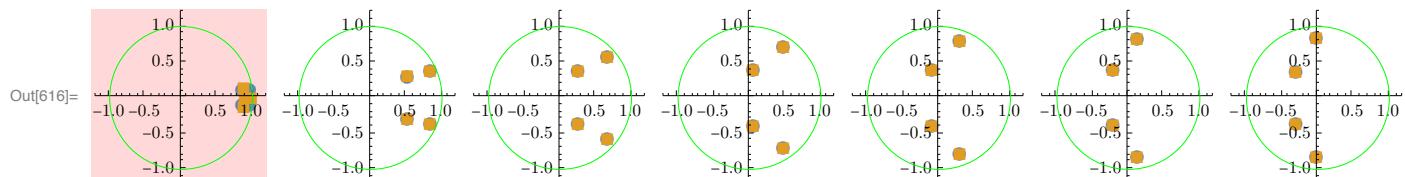
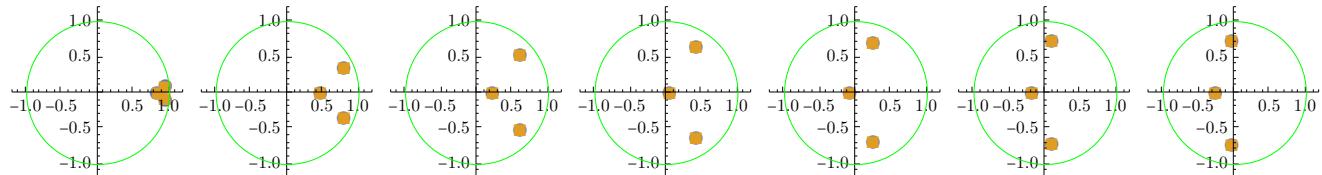
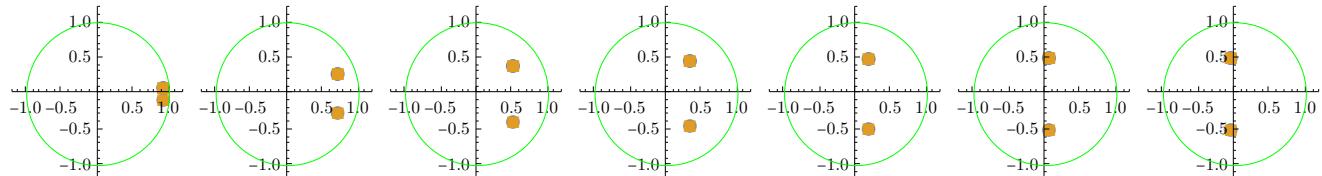


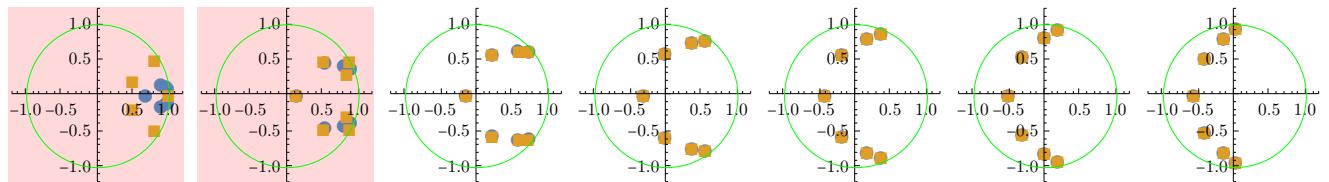


★ Chebyshev 2

```
In[616]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

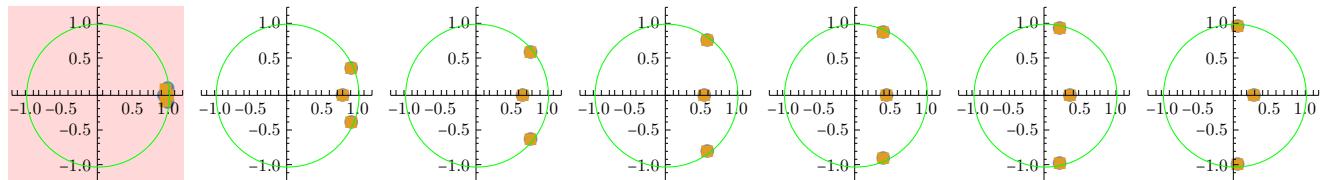
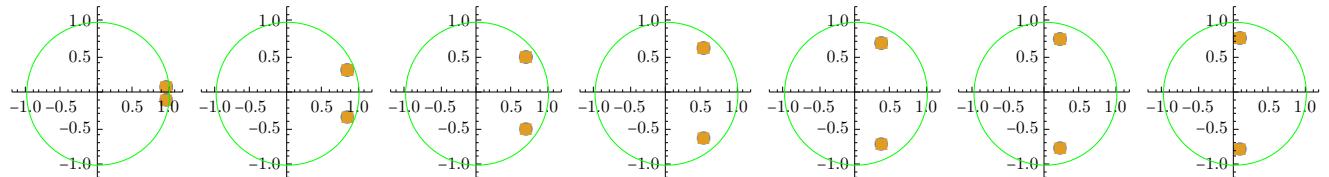
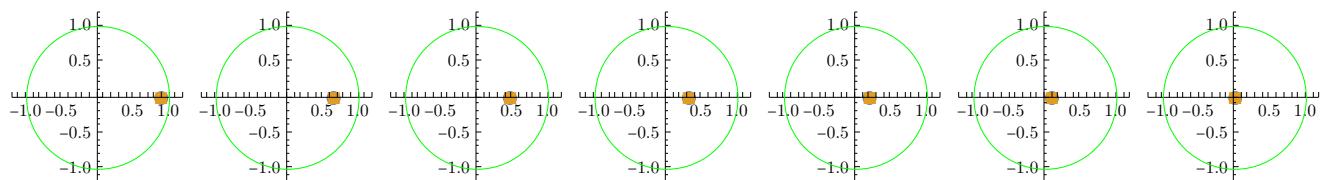


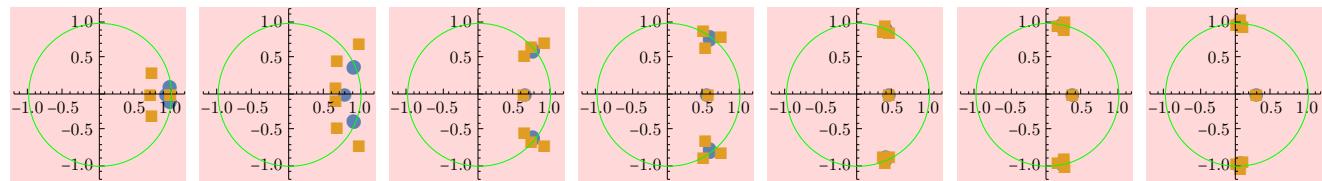
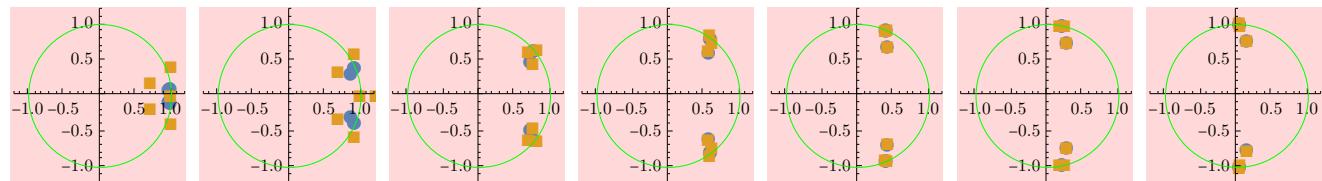
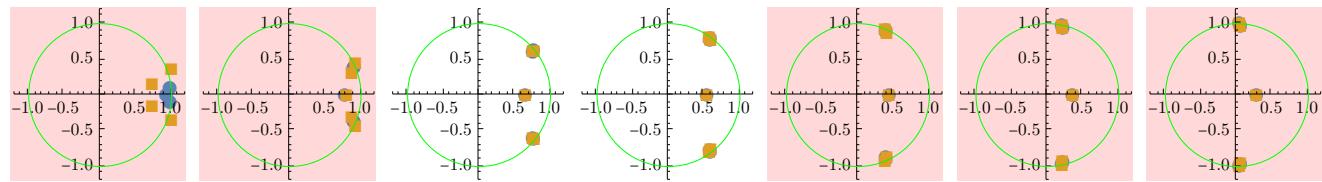
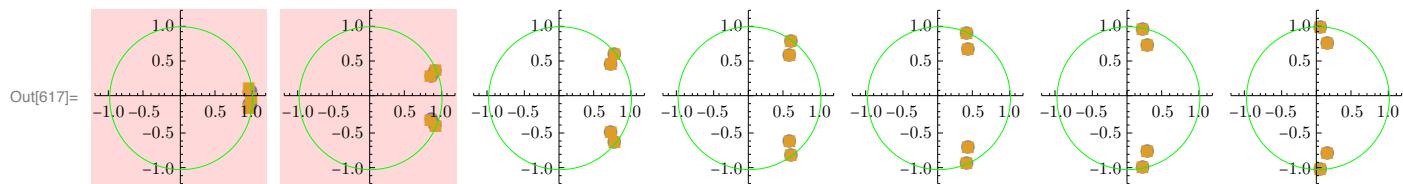




★ Eliptyczne

```
In[617]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





21.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

21.2.1 Dyskretyzacja

```
In[618]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[619]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[620]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

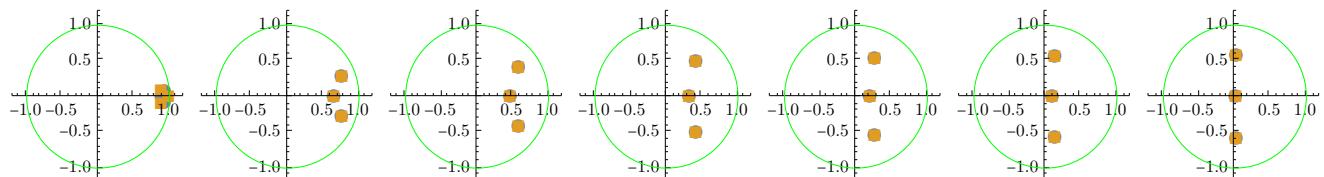
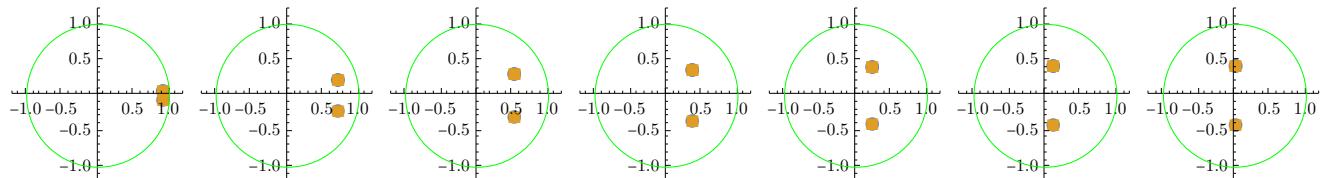
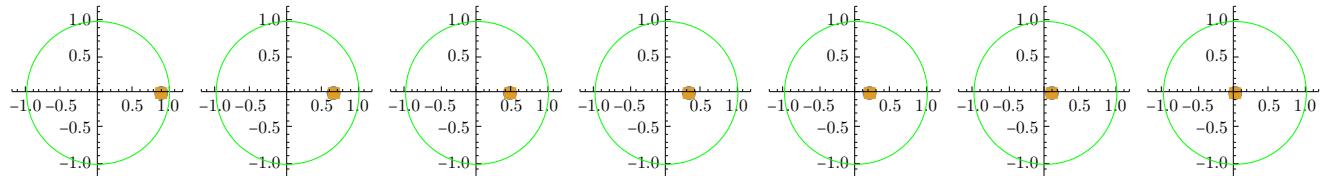
```
In[621]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

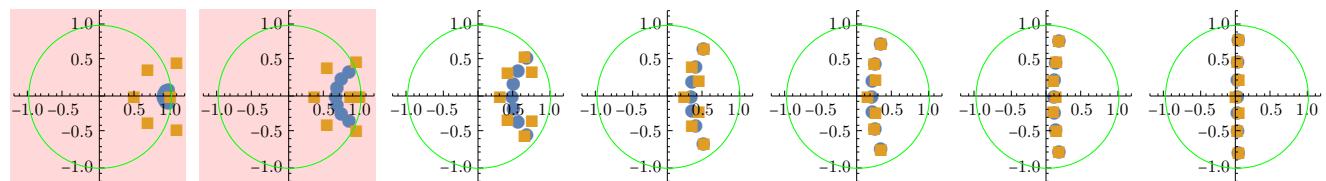
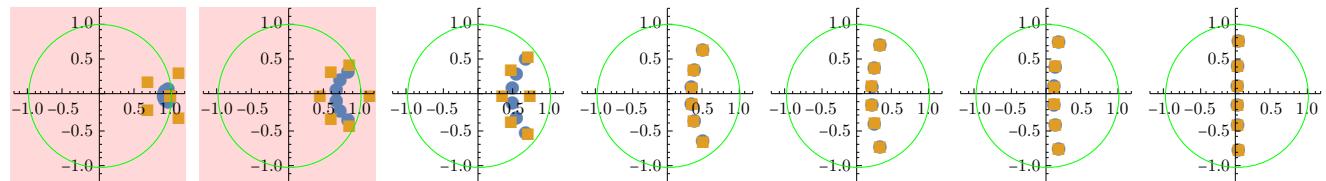
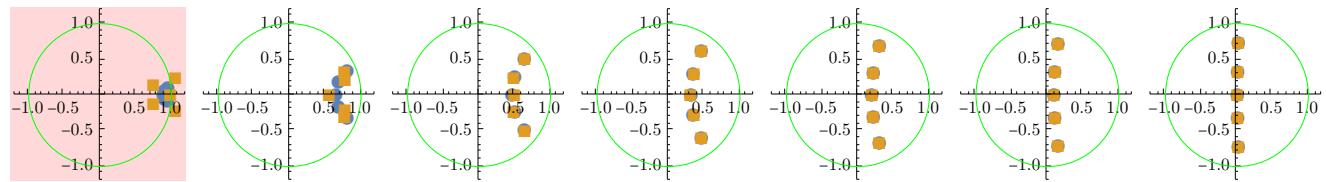
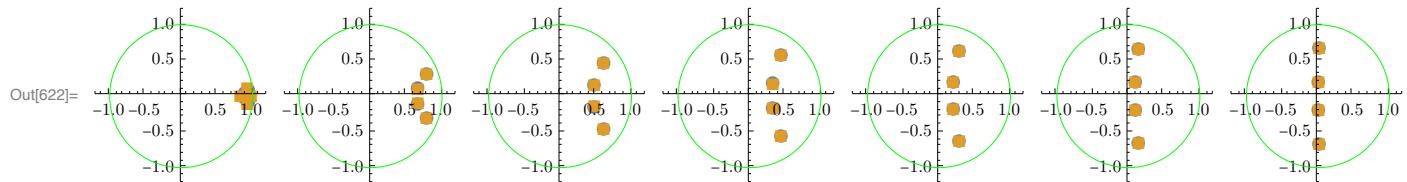
21.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

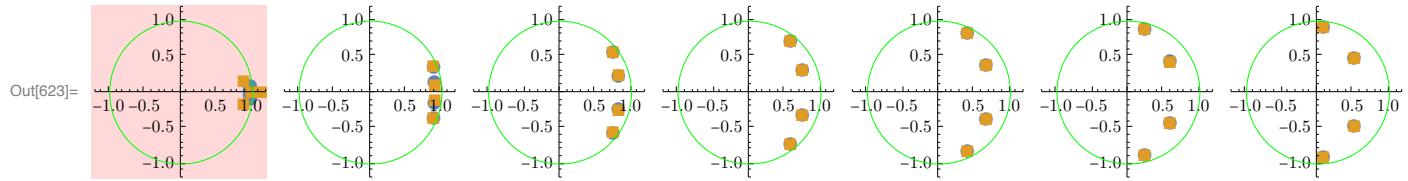
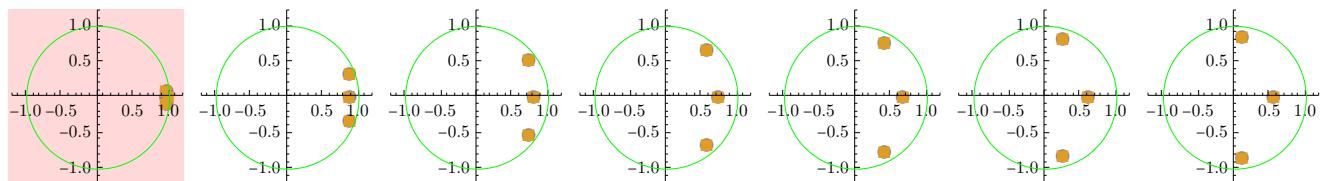
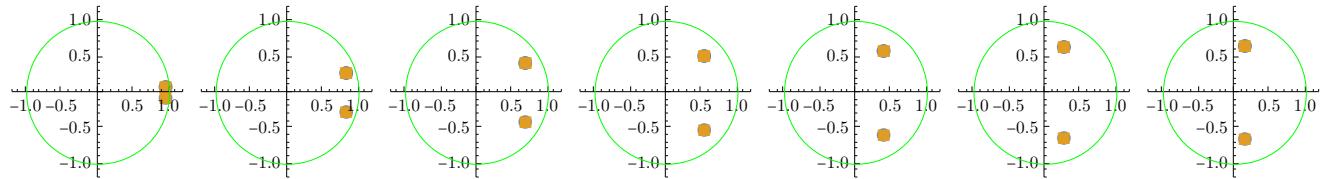
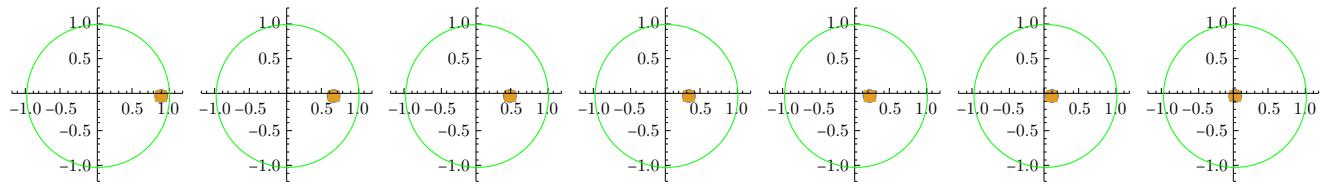
```
In[622]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

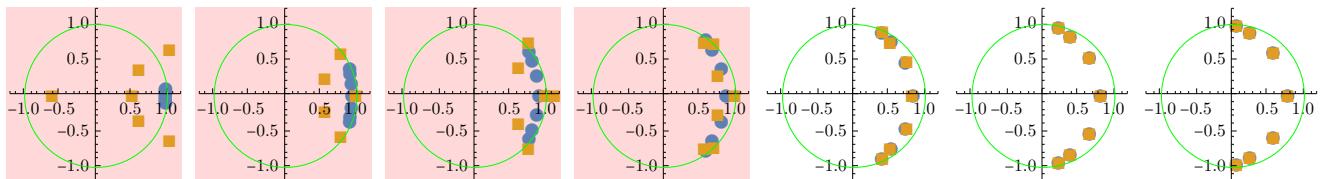
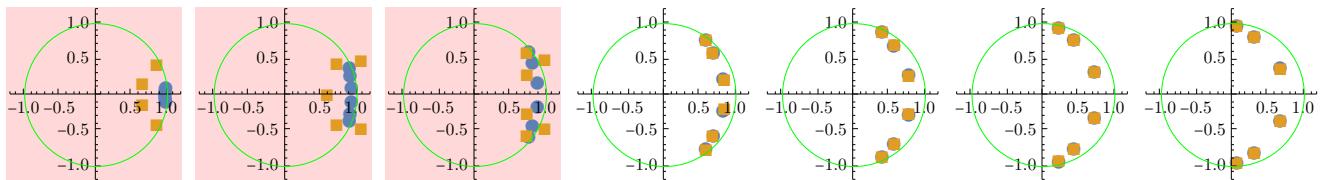
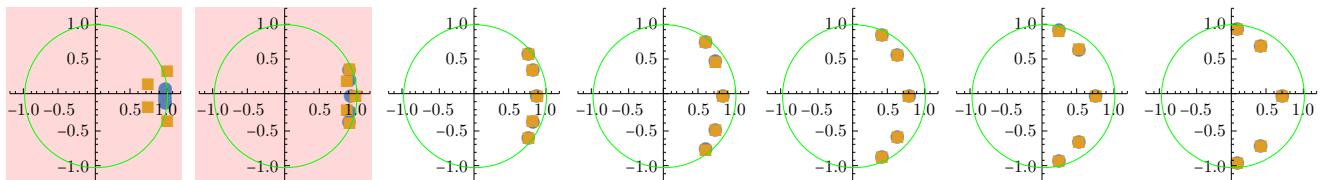




★ Chebyshev 1

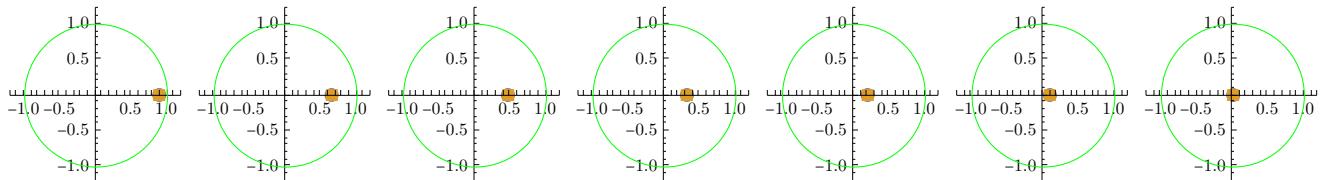
```
In[623]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

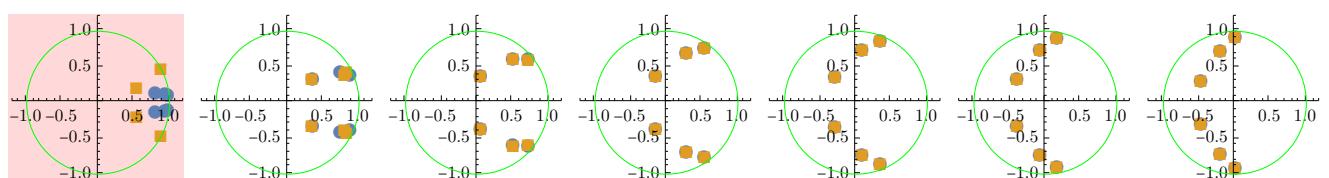
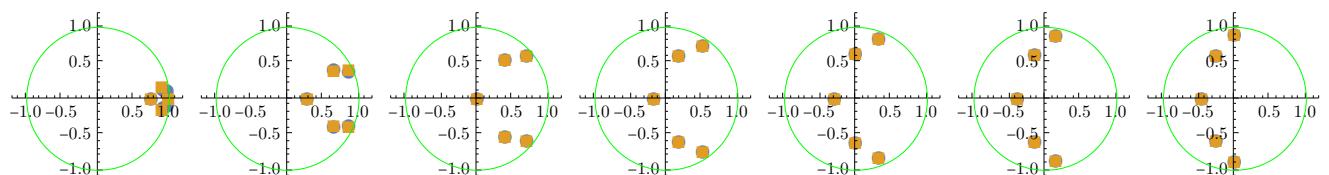
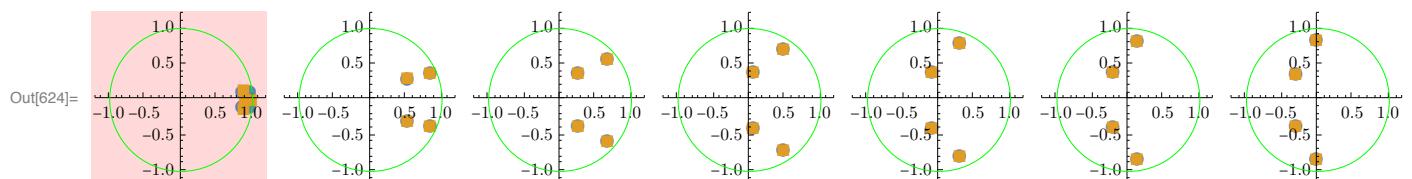
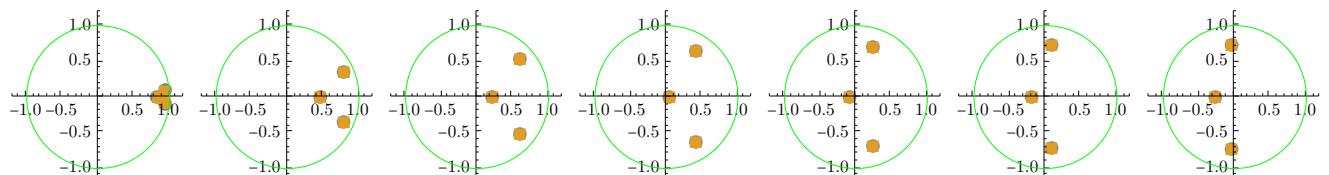
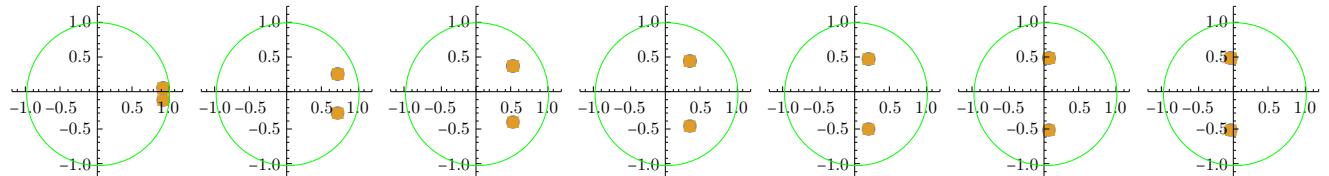


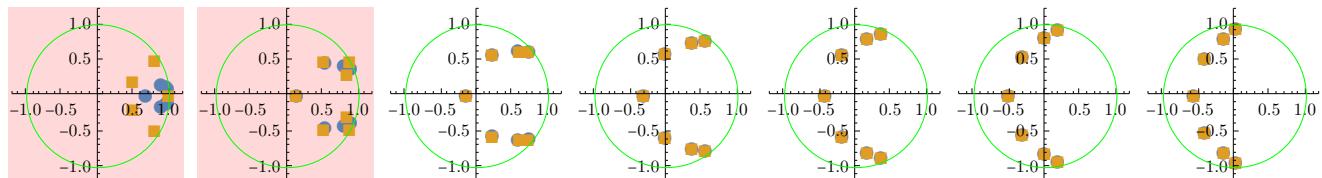


★ Chebyshev 2

```
In[624]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

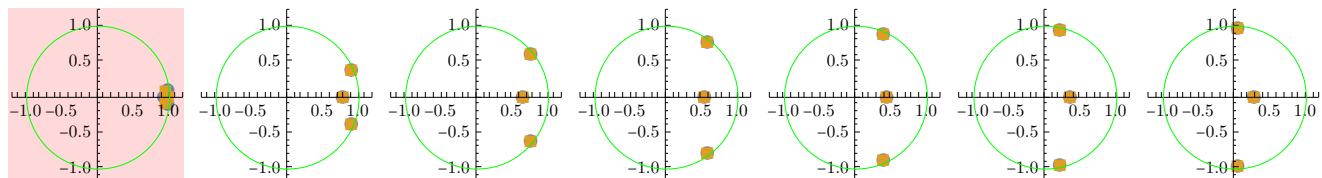
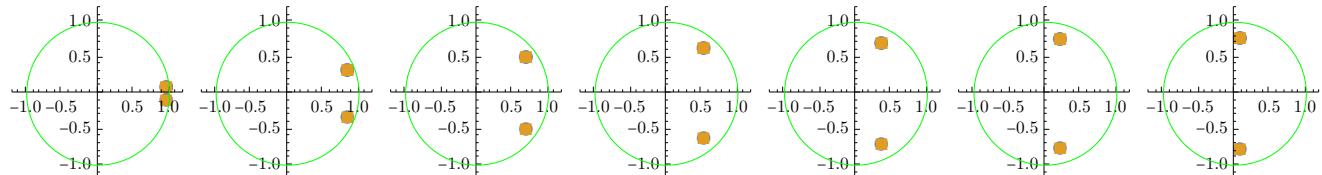
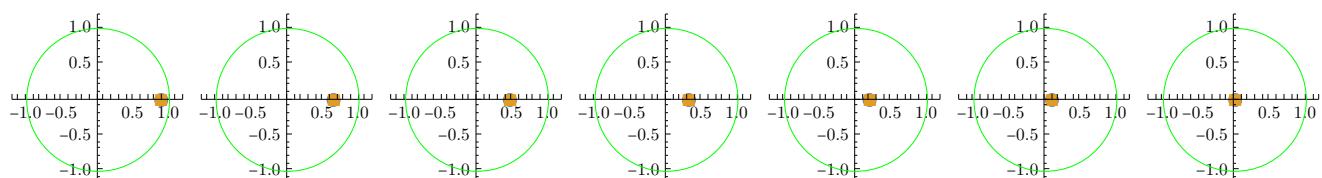


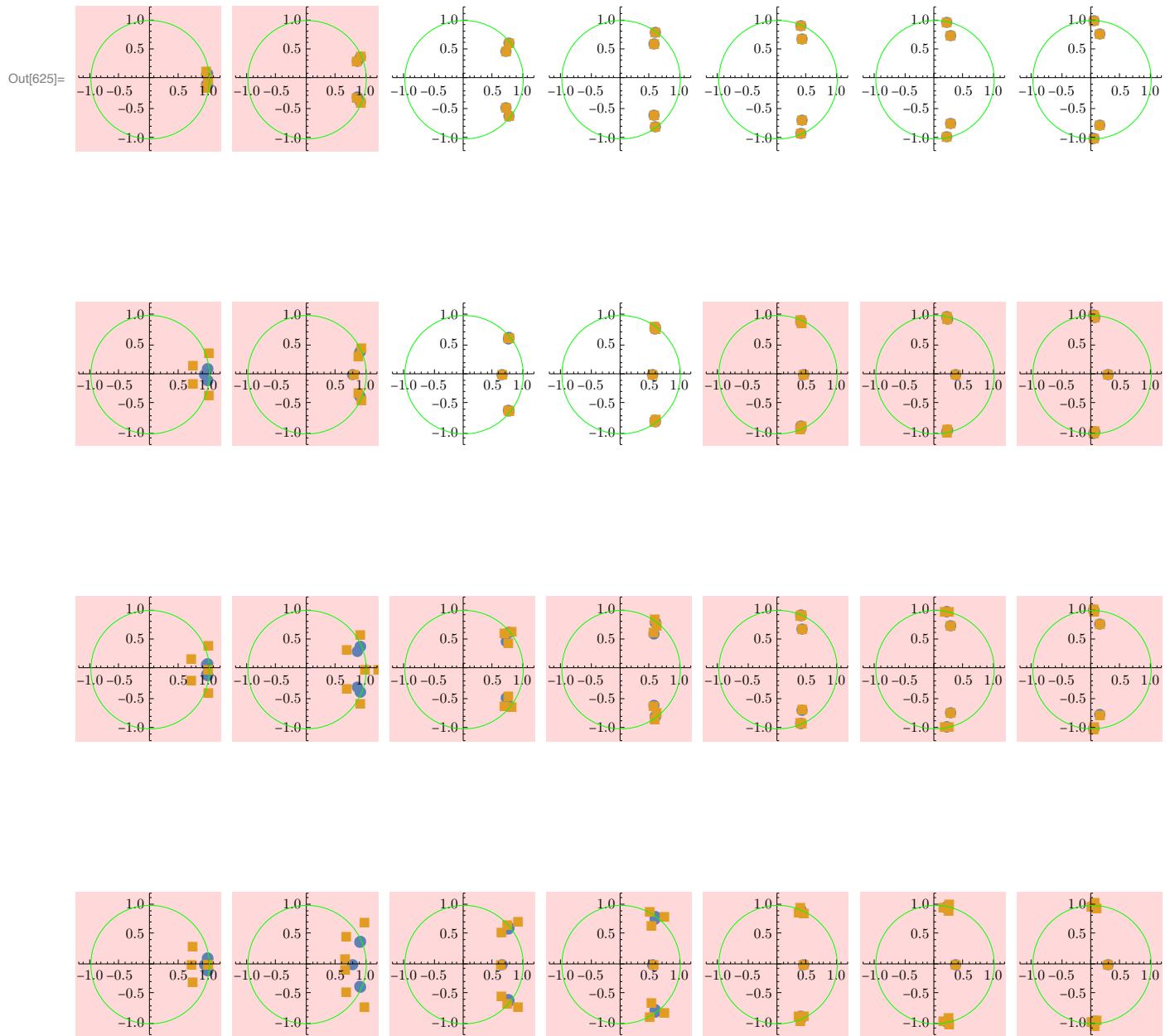




★ Eliptyczne

```
In[625]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





22 Badanie stabilności filtrów cyfrowych (13 bit)

```
In[643]:= bity = 13;
```

22.1 Dyskretyzacja na poziomie współczynników

22.1.1 Dyskretyzacja

```
In[644]:= DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];
```

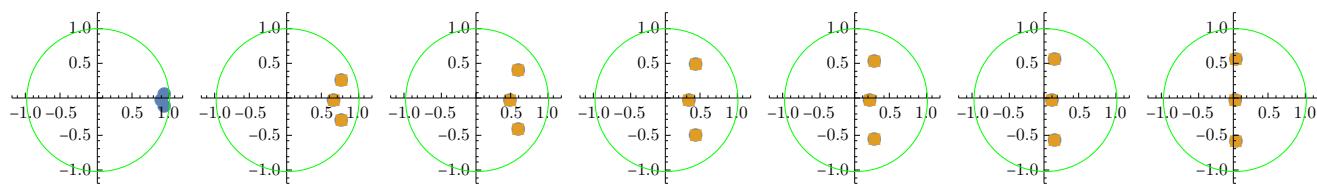
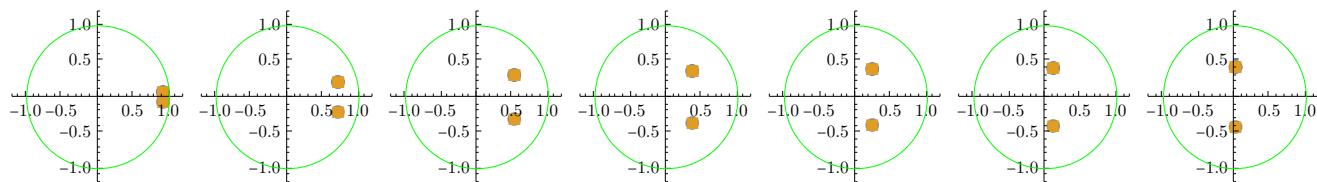
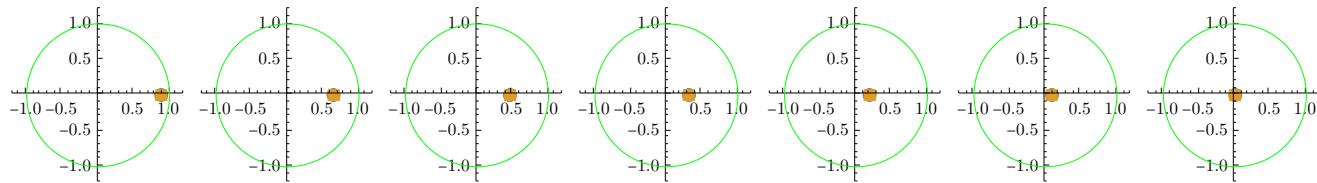
```
In[645]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[646]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[647]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

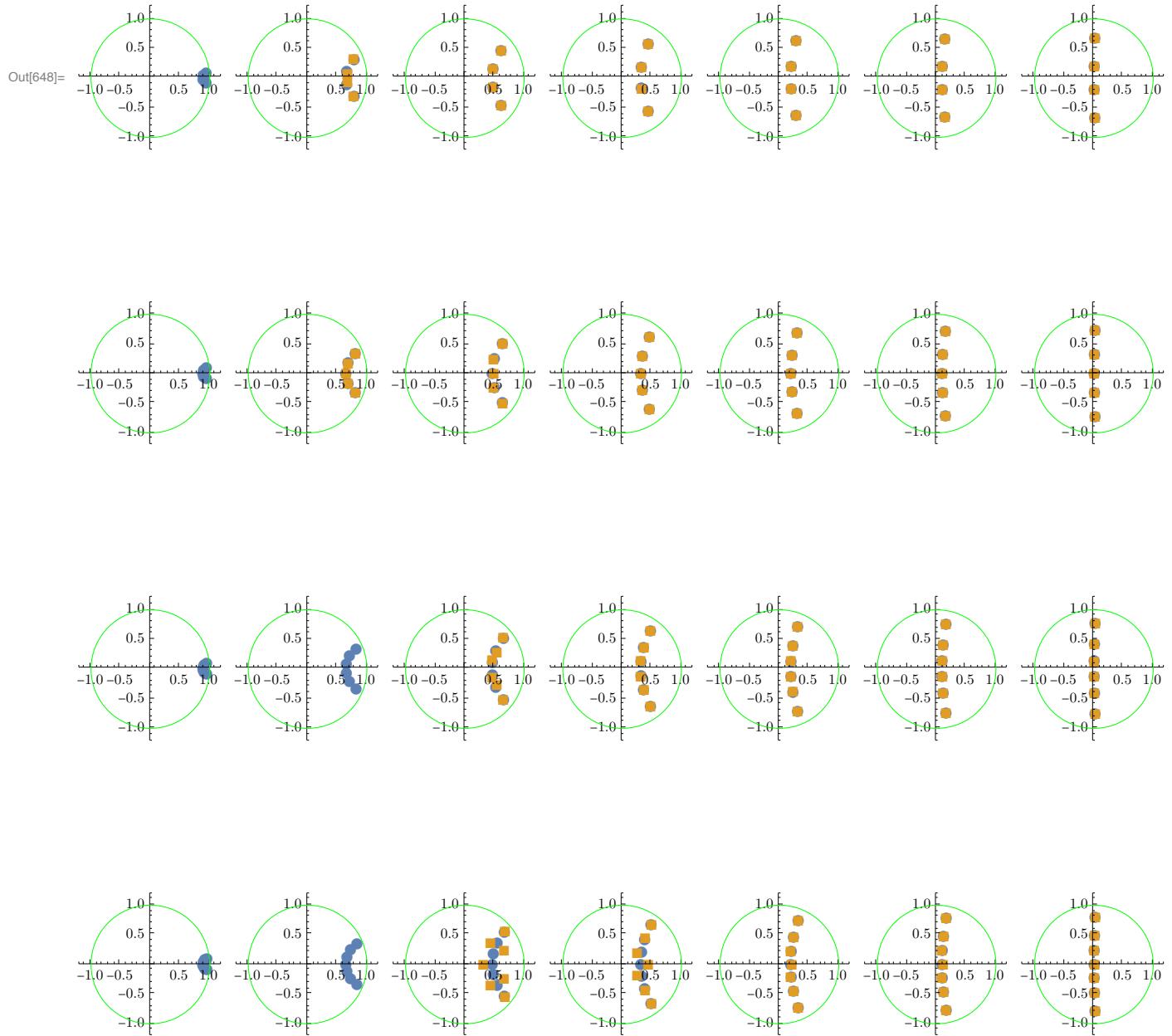
22.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

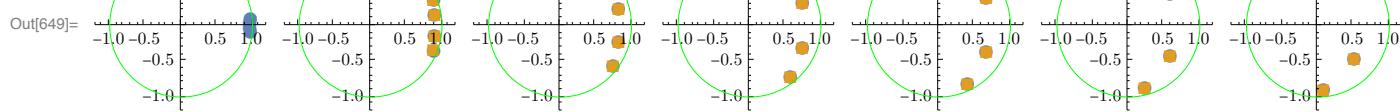
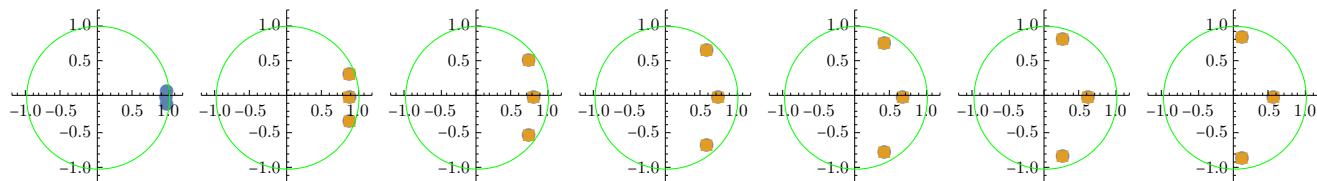
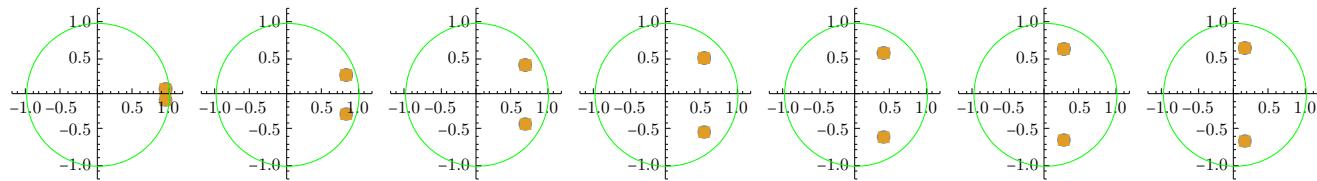
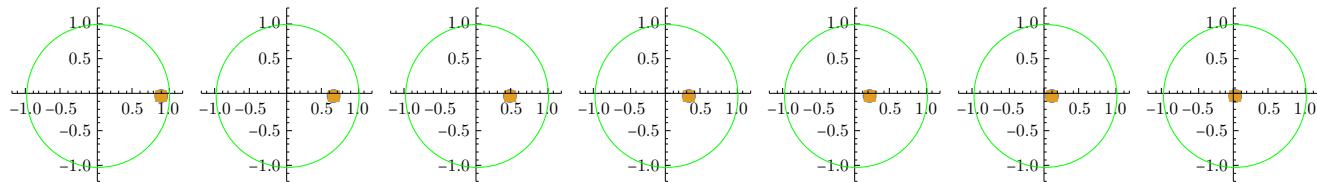
```
In[648]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

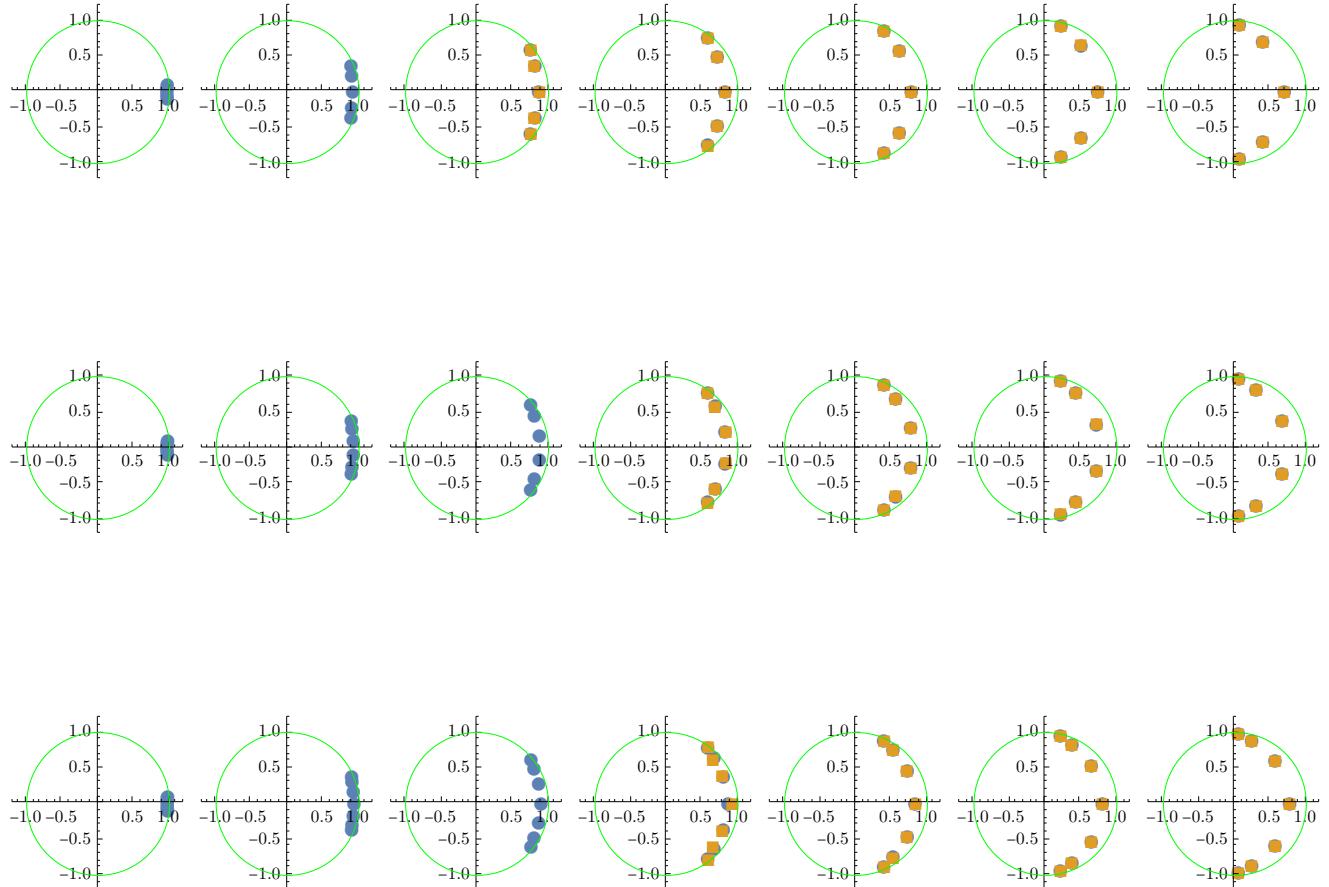




★ Chebyshev 1

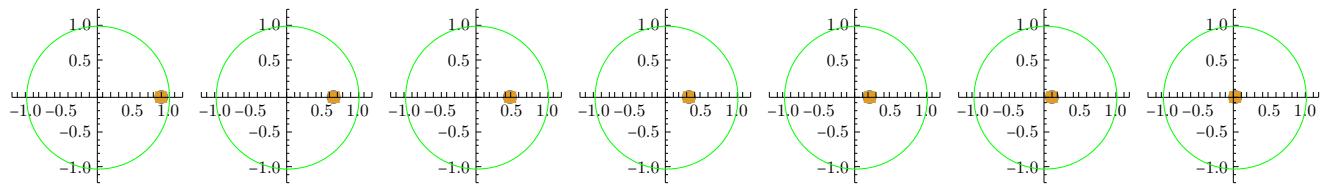
```
In[649]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

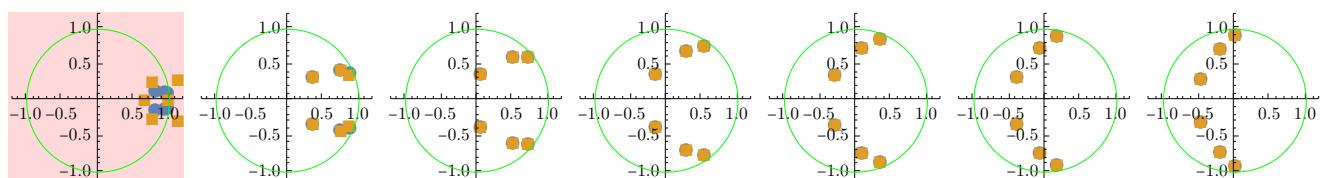
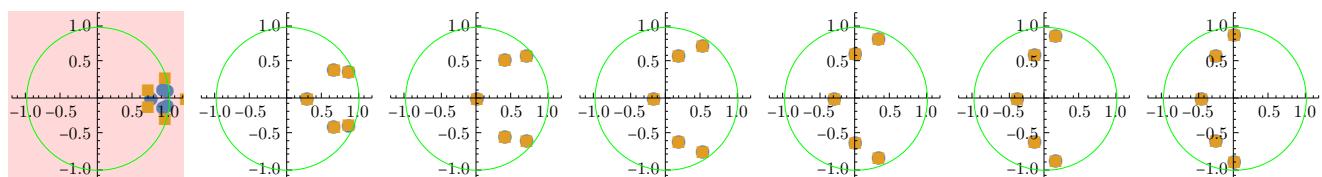
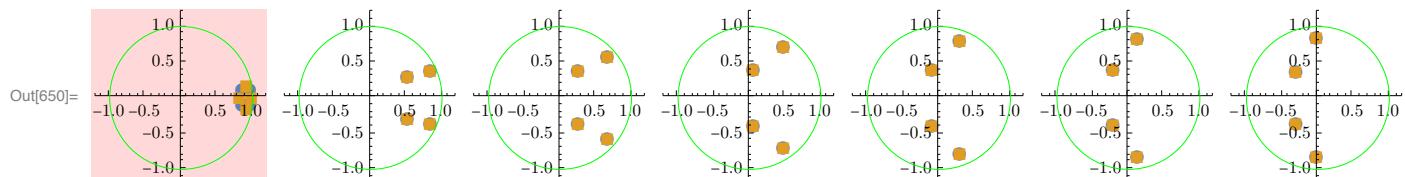
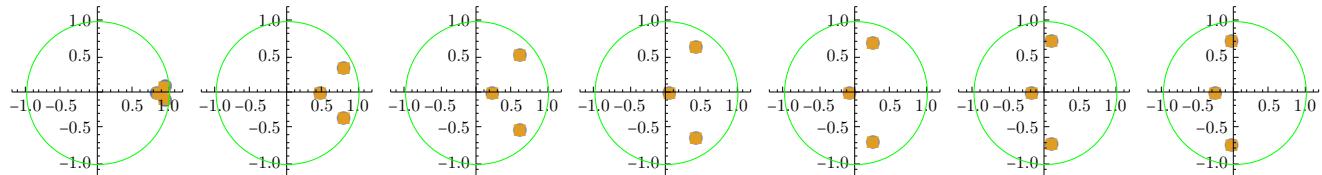
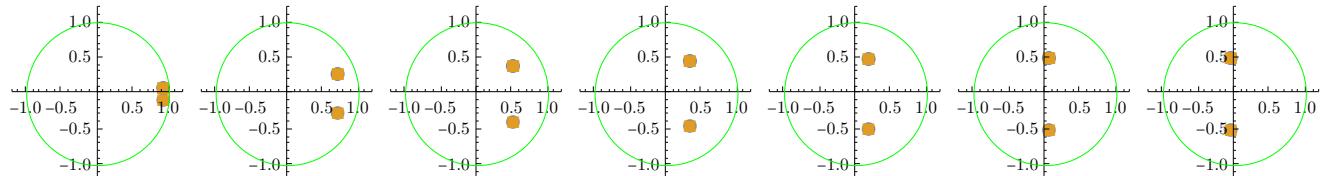


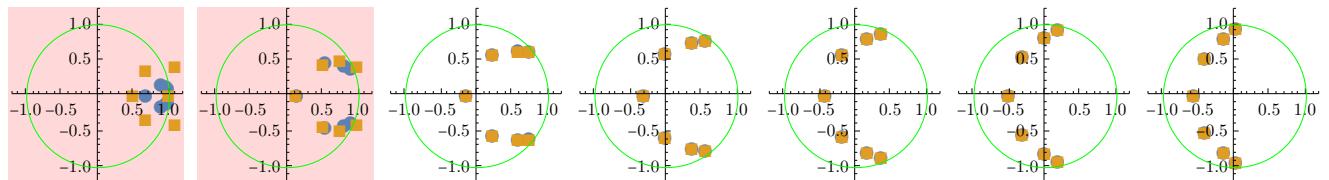


★ Chebyshev 2

```
In[650]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

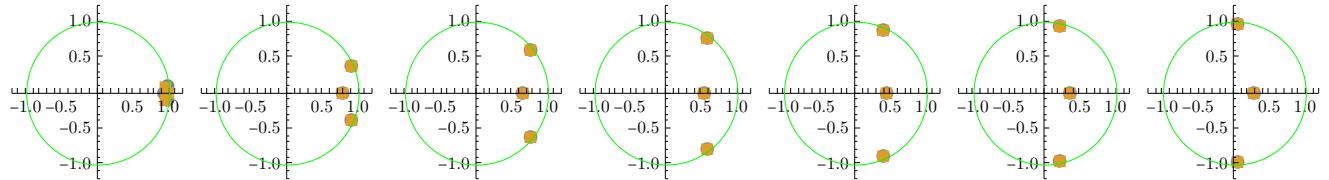
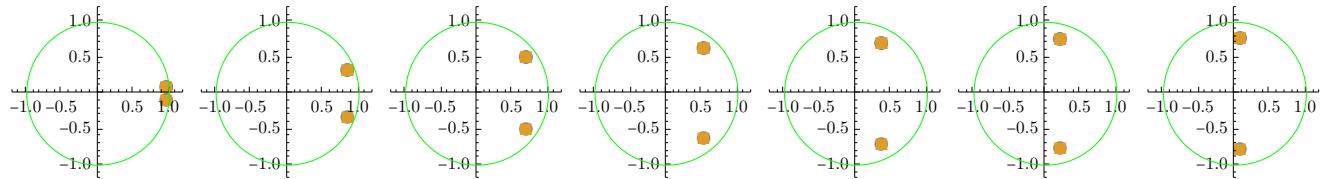
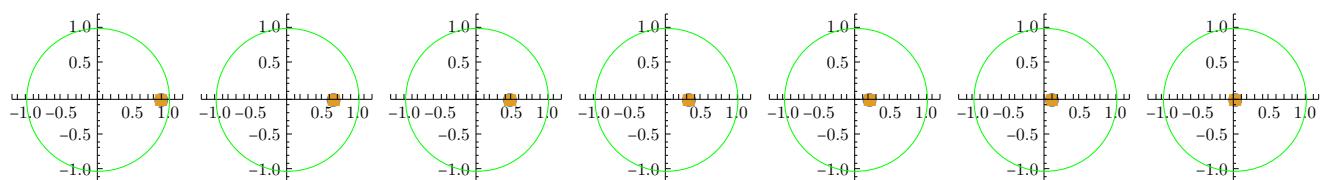


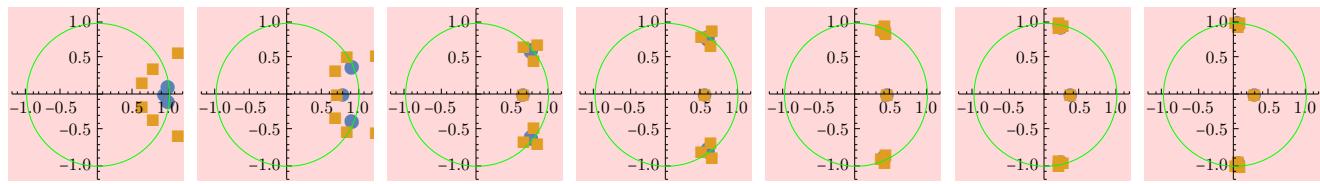
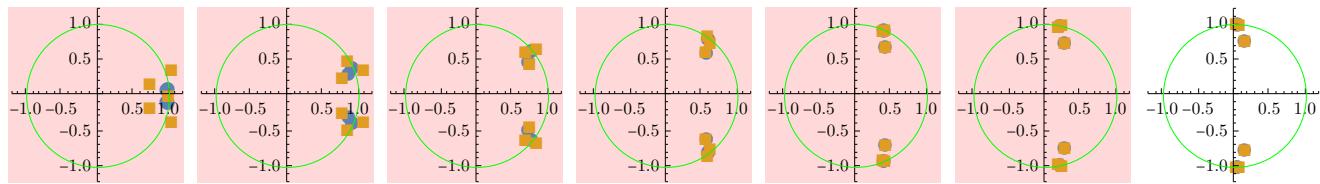
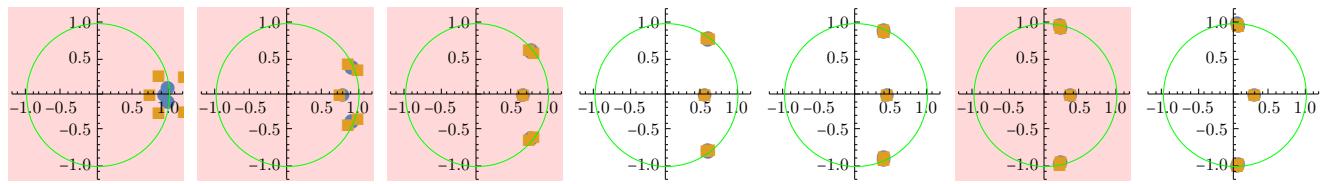
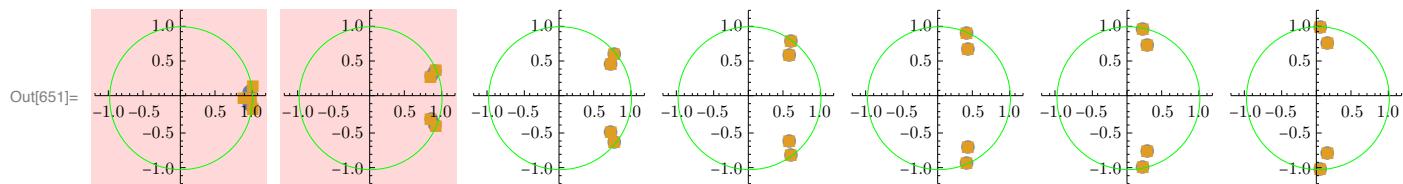




★ Eliptyczne

```
In[651]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





22.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

22.2.1 Dyskretyzacja

```
In[652]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[653]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[654]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

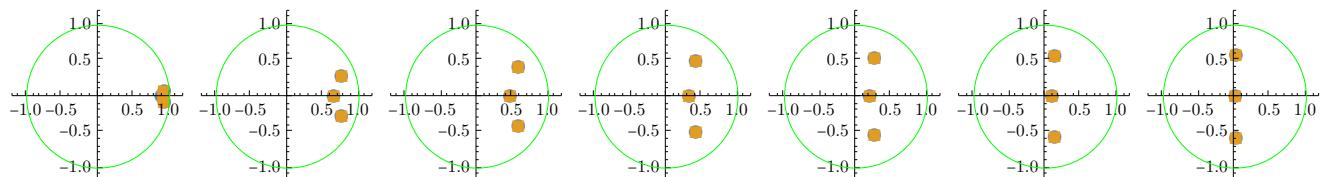
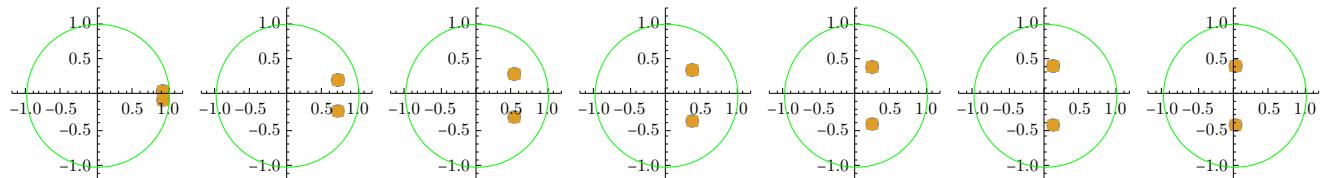
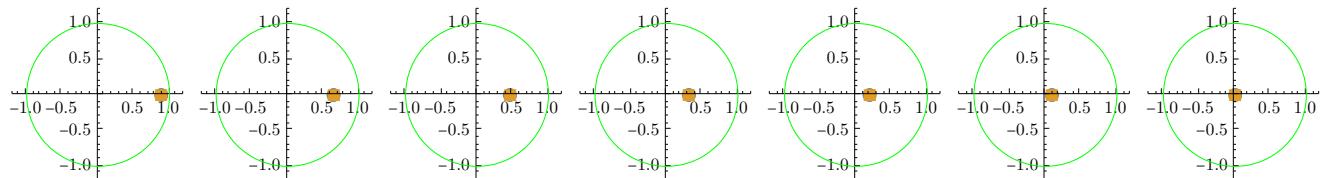
```
In[655]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

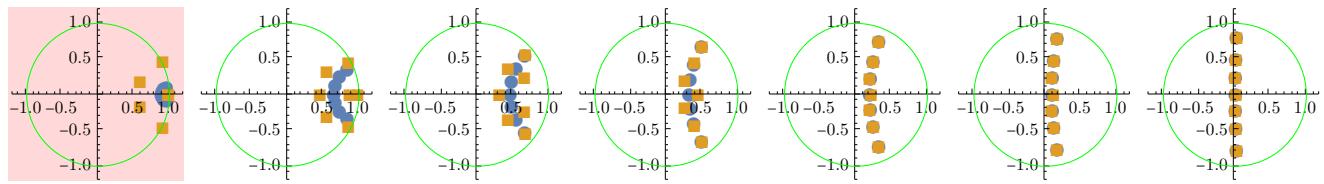
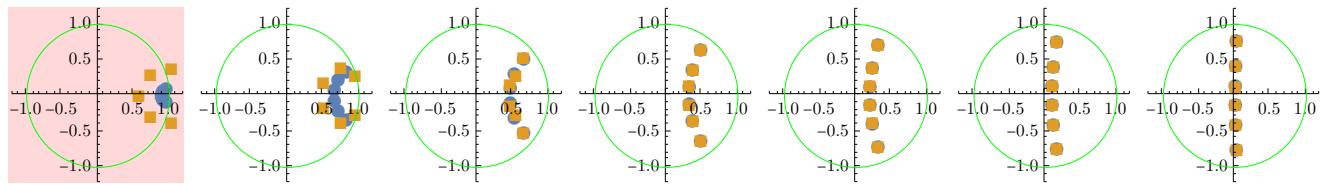
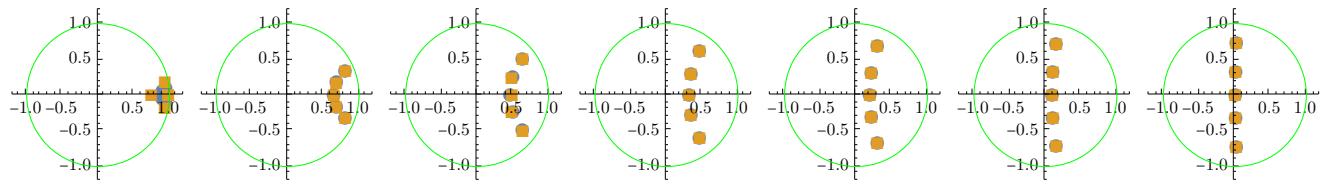
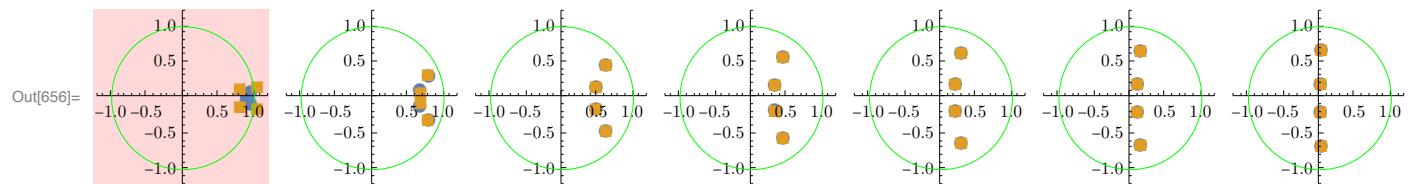
22.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

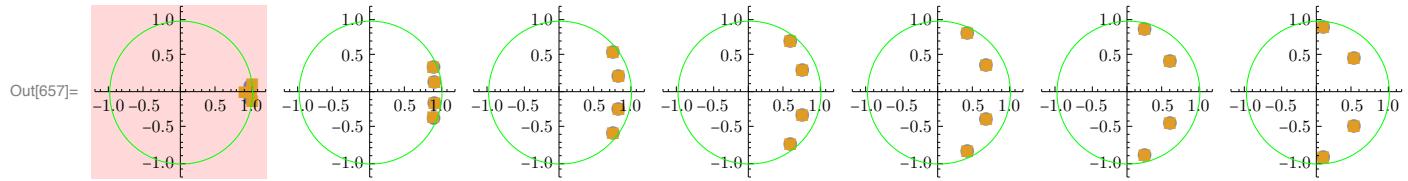
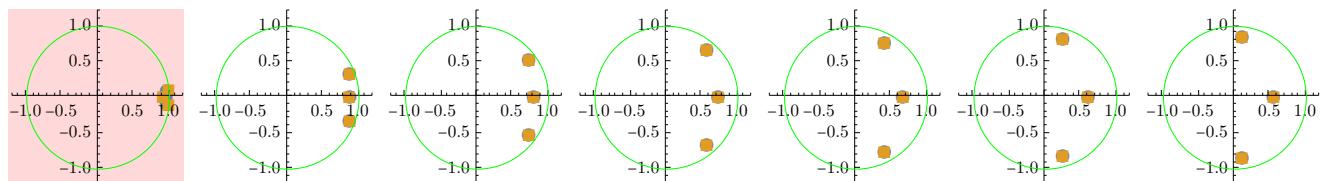
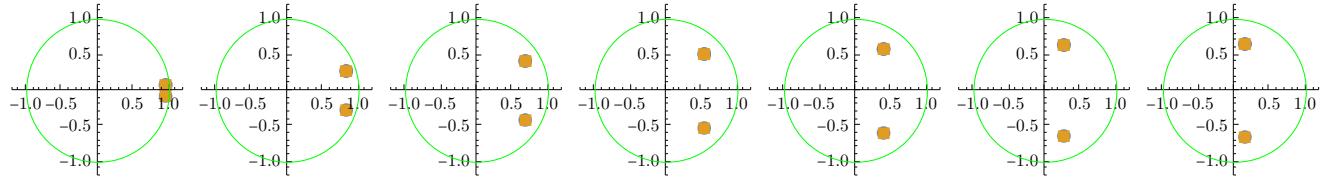
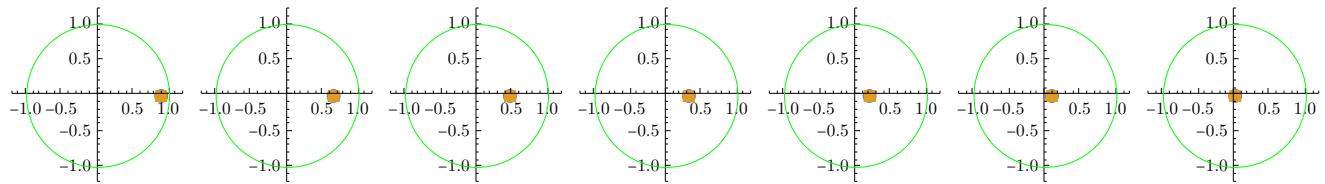
```
In[656]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

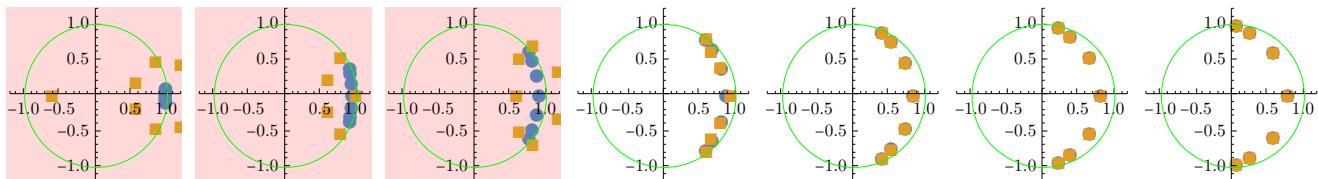
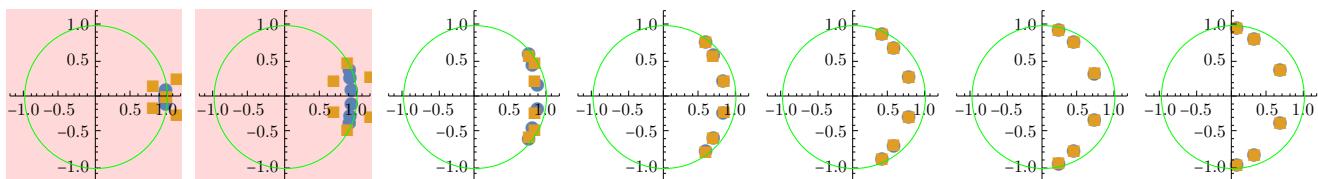
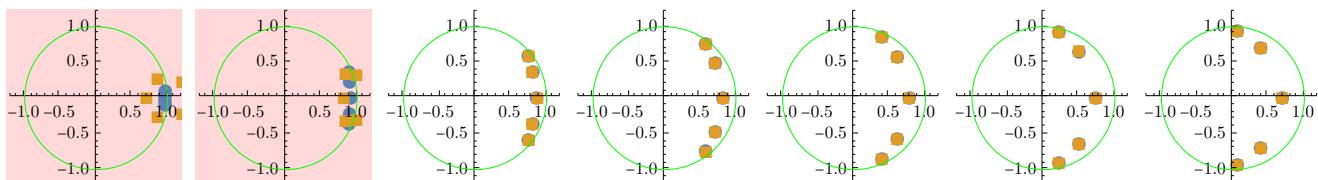




★ Chebyshev 1

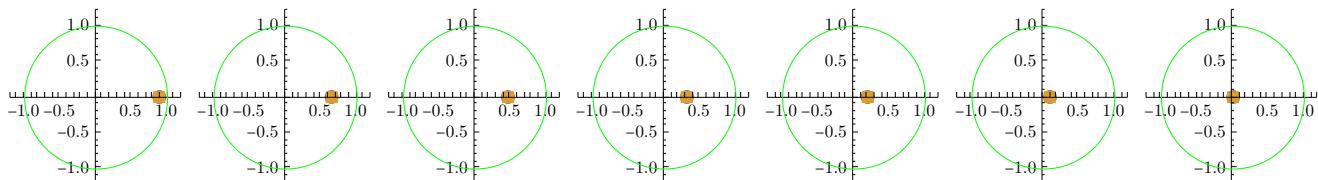
```
In[657]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

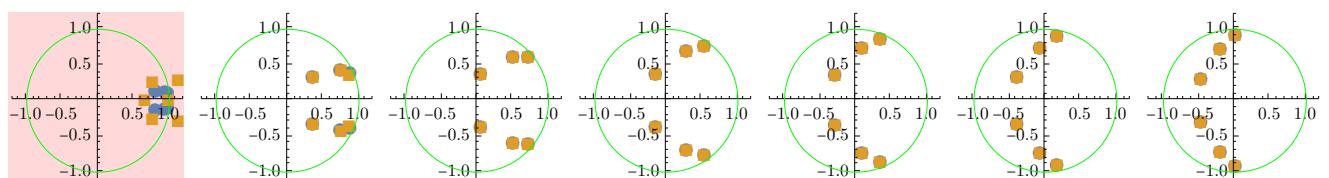
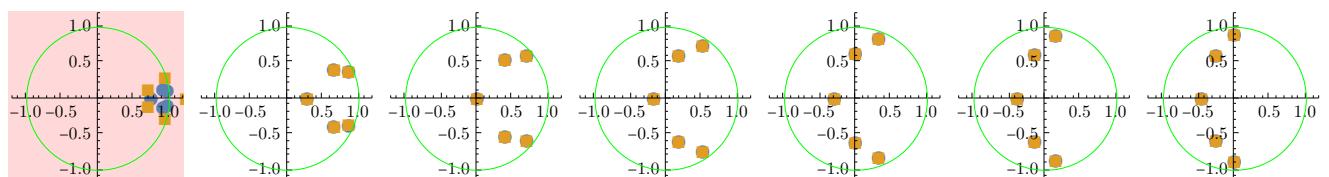
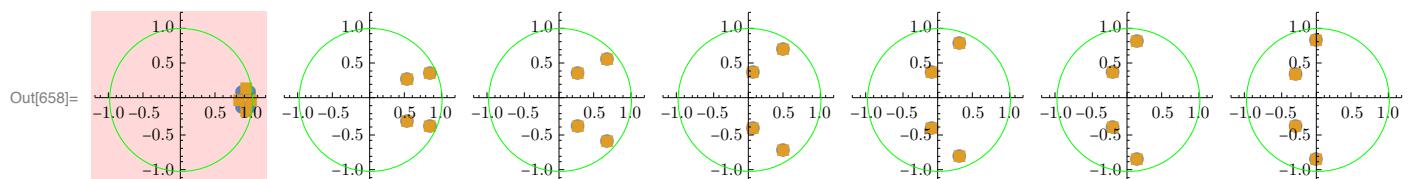
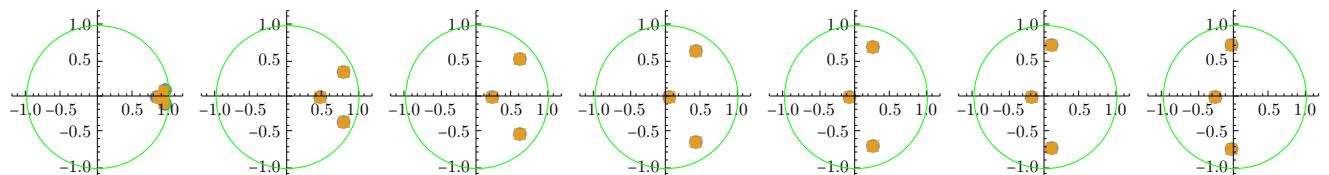
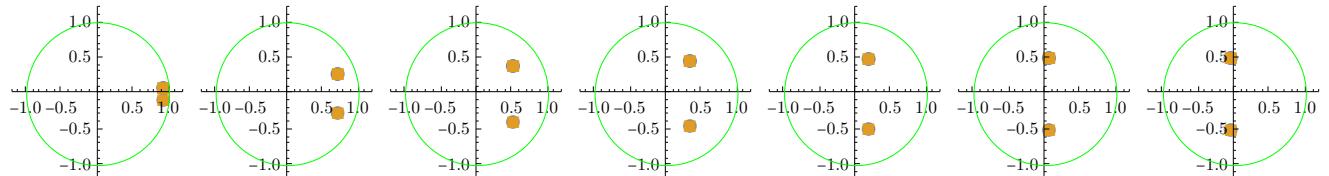


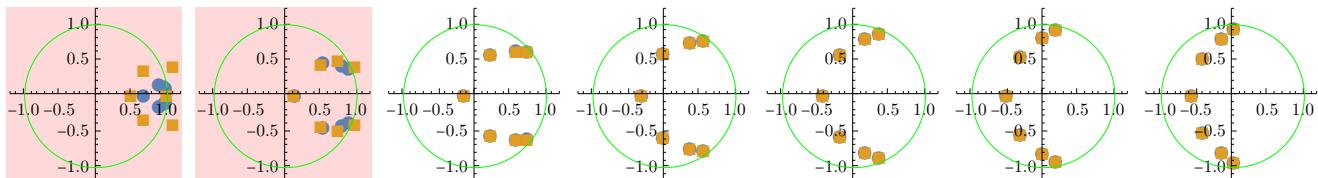


★ Chebyshev 2

```
In[658]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

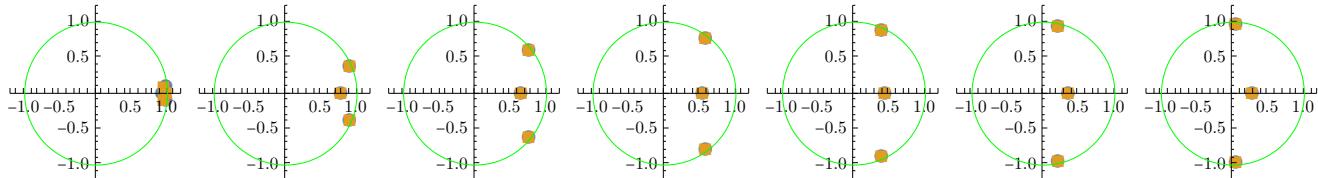
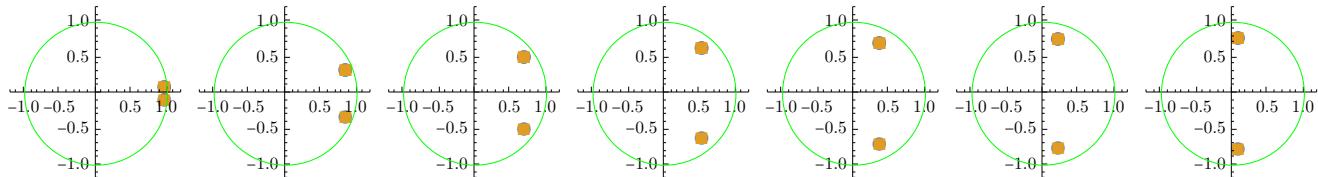
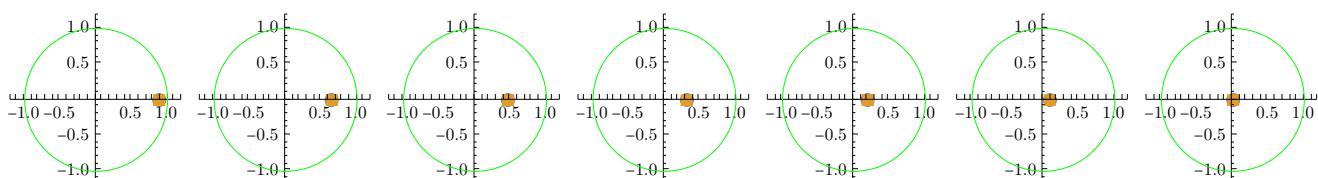


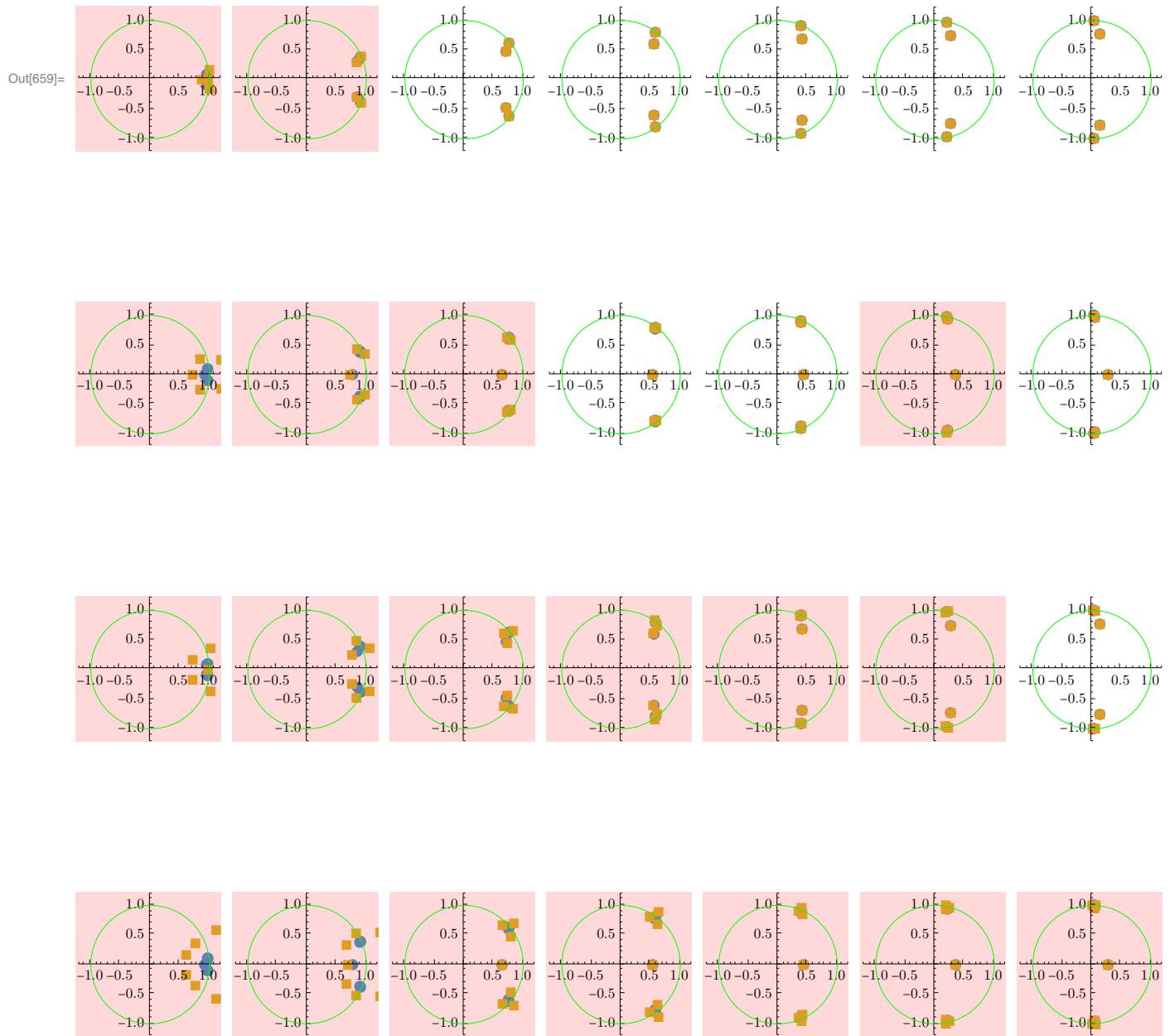




★ Eliptyczne

```
In[659]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





23 Badanie stabilności filtrów cyfrowych (14 bit)

```
In[660]:= bity = 14;
```

23.1 Dyskretyzacja na poziomie współczynników

23.1.1 Dyskretyzacja

```
In[661]:= DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];
```

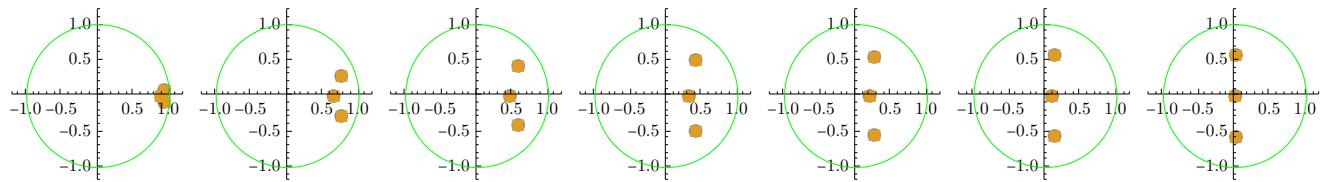
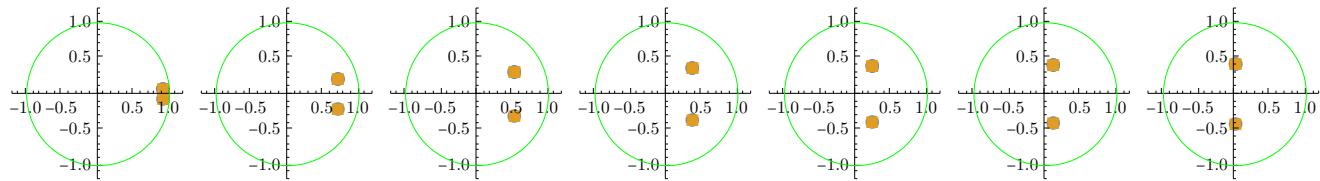
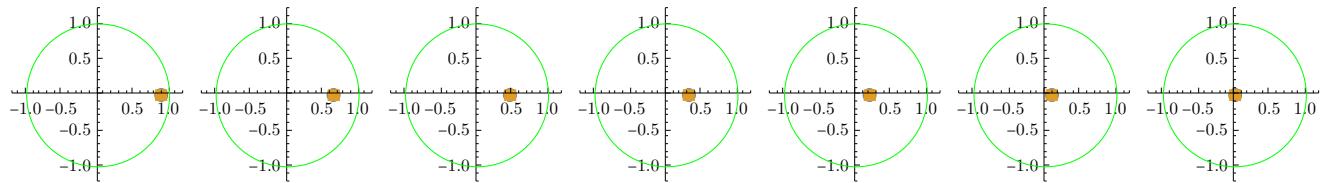
```
In[662]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[663]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[664]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

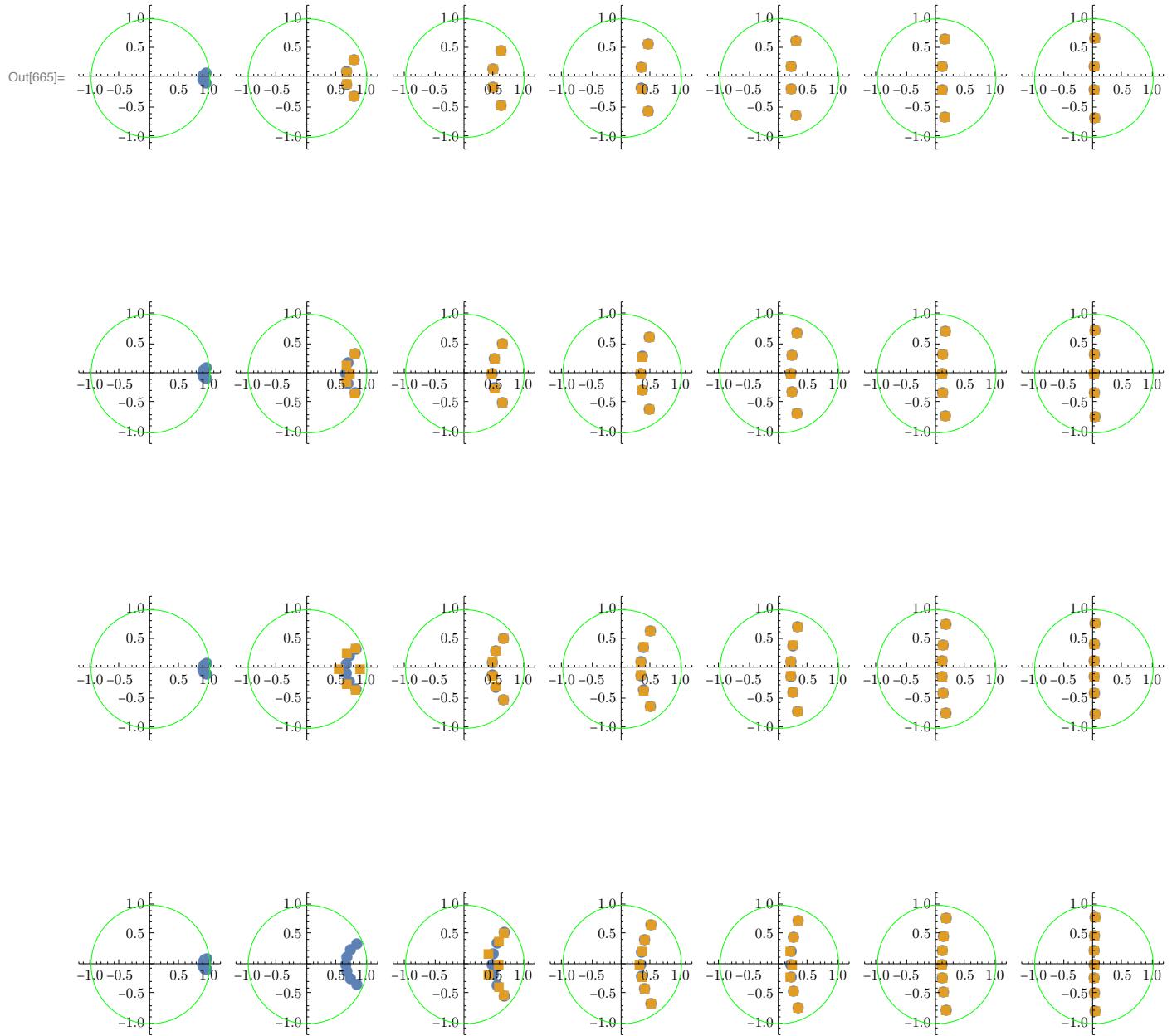
23.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

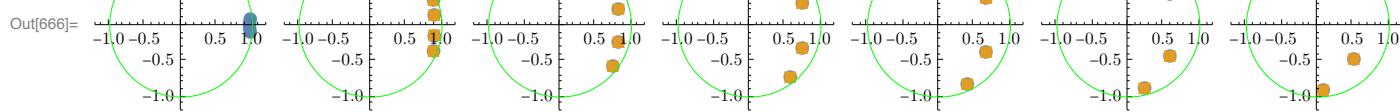
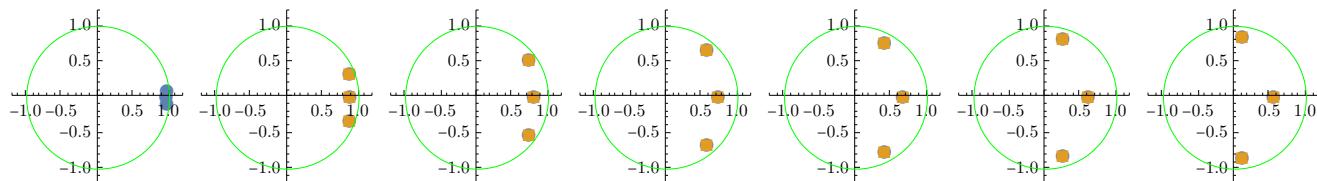
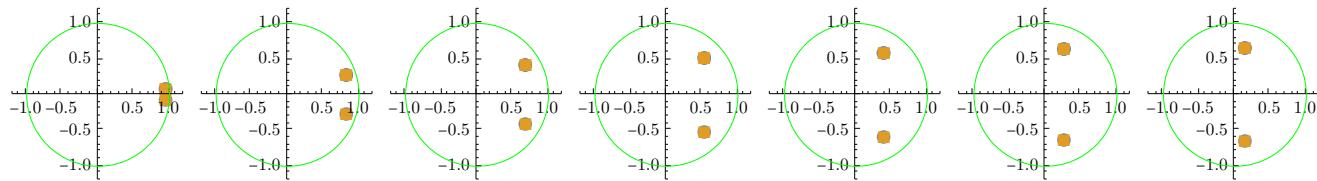
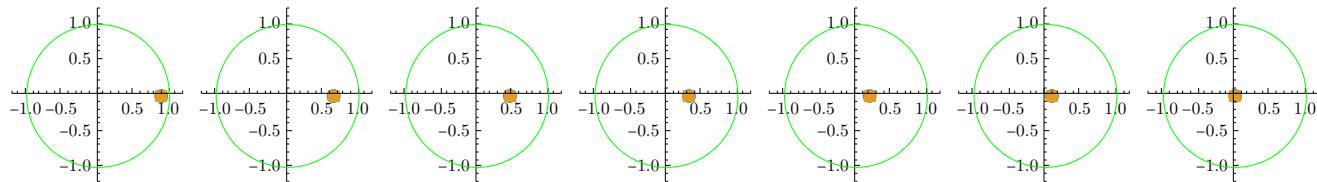
```
In[665]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

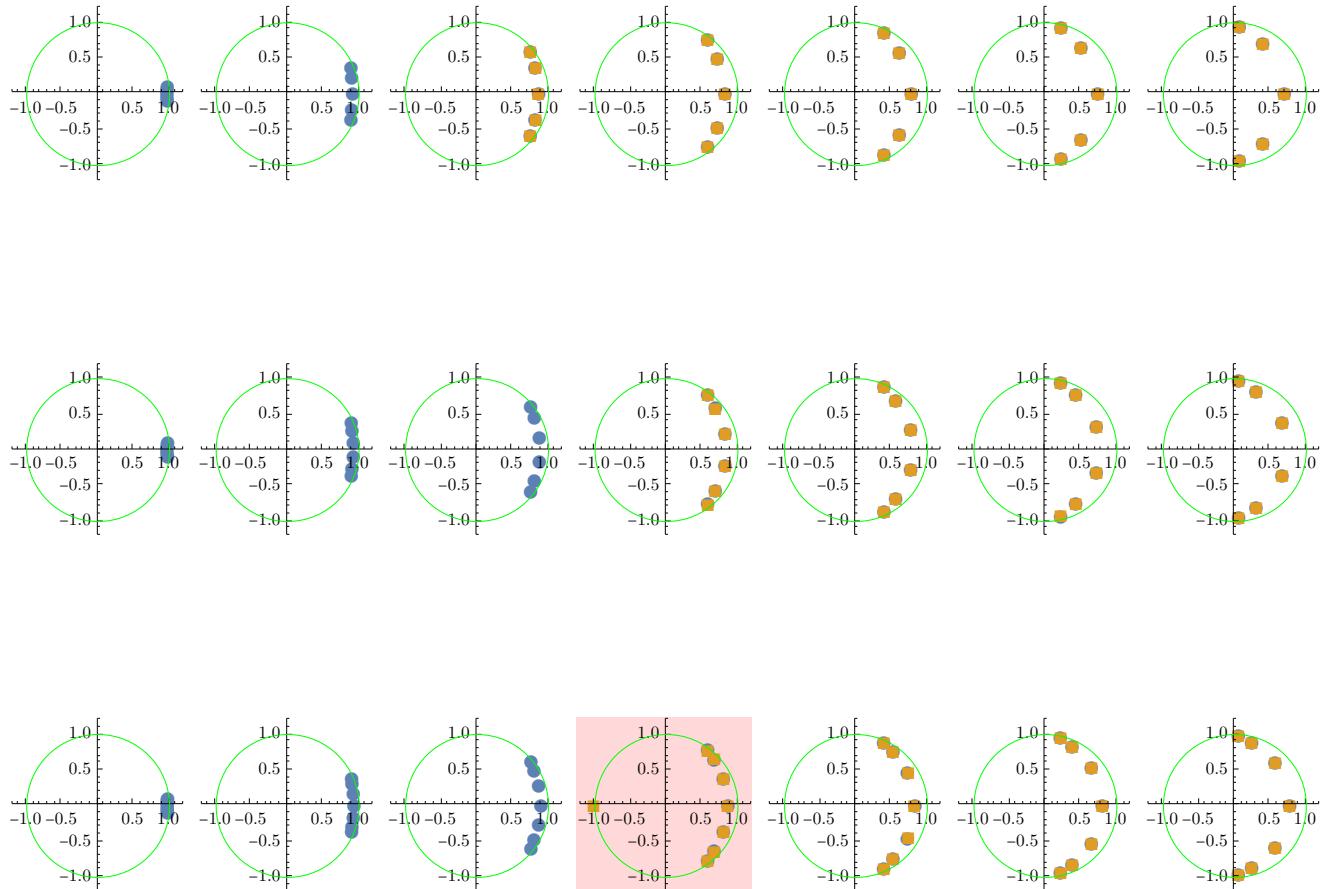




★ Chebyshev 1

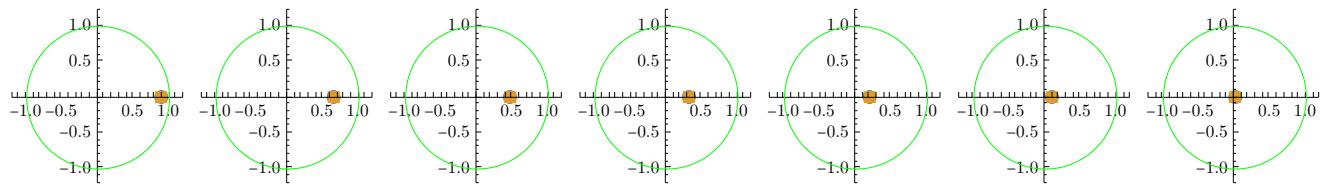
```
In[666]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

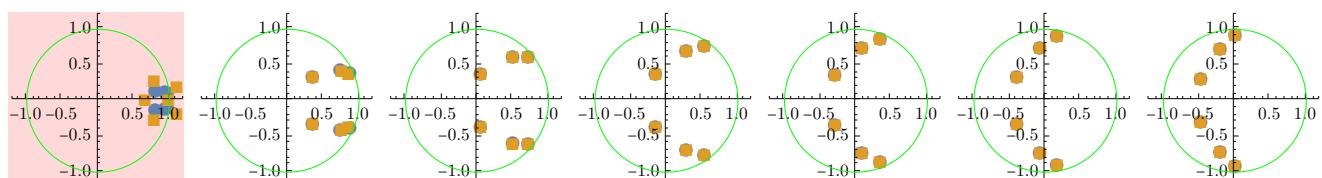
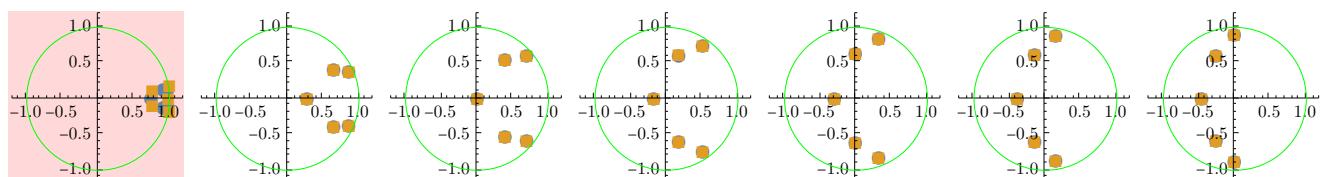
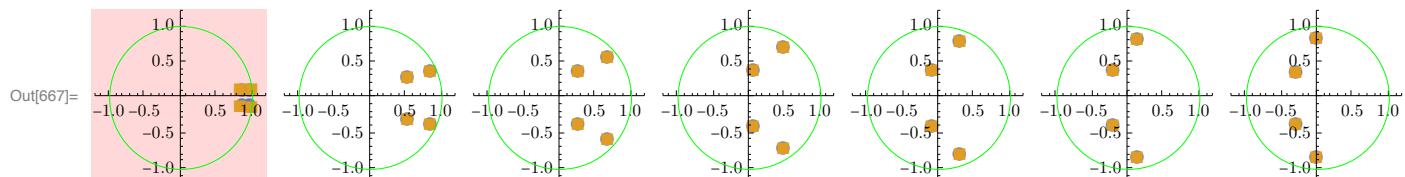
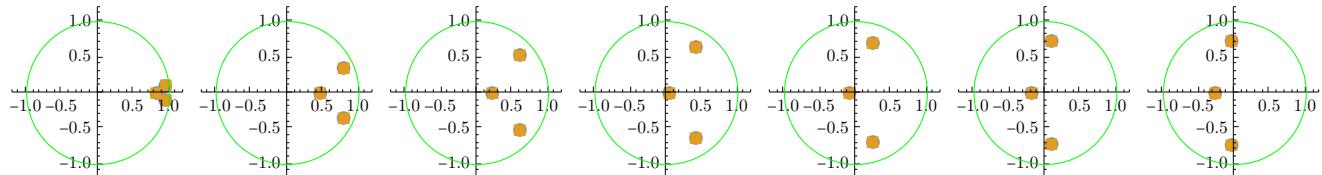
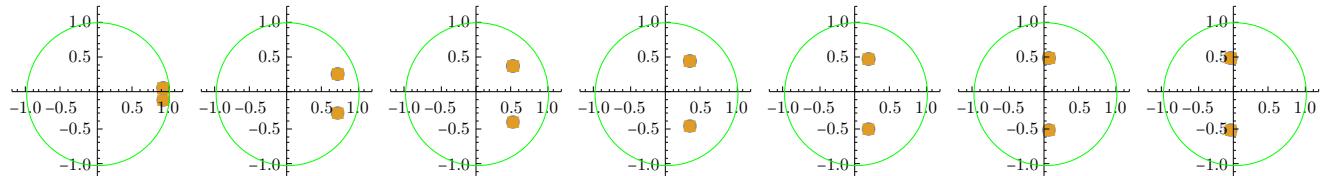


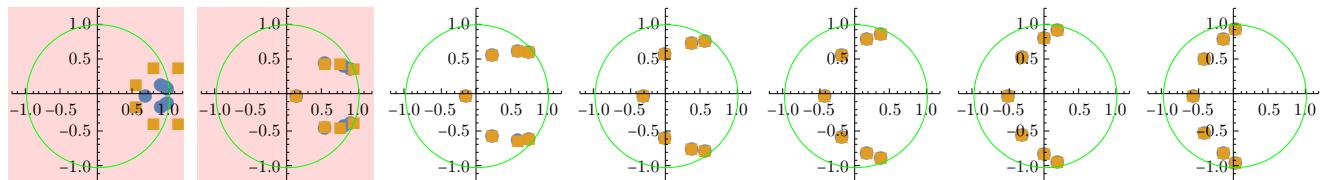


★ Chebyshev 2

```
In[667]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

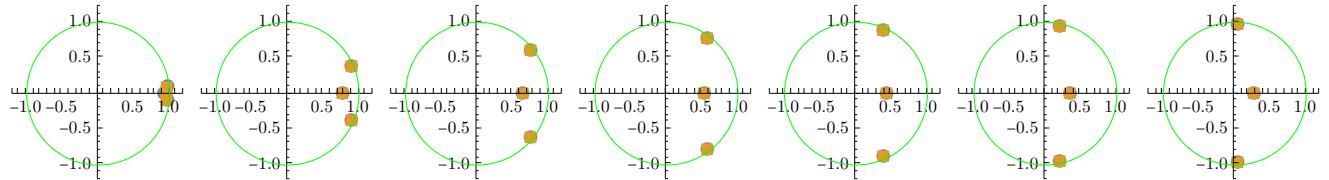
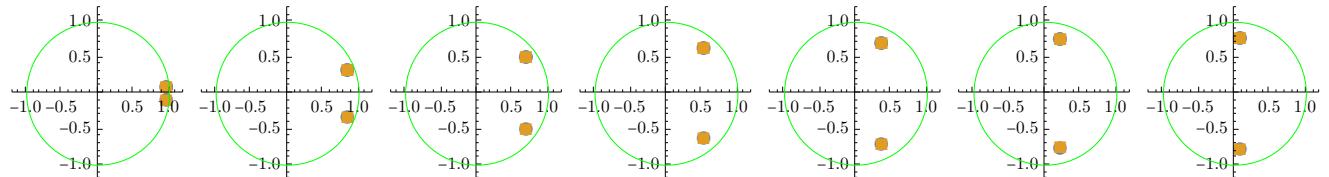
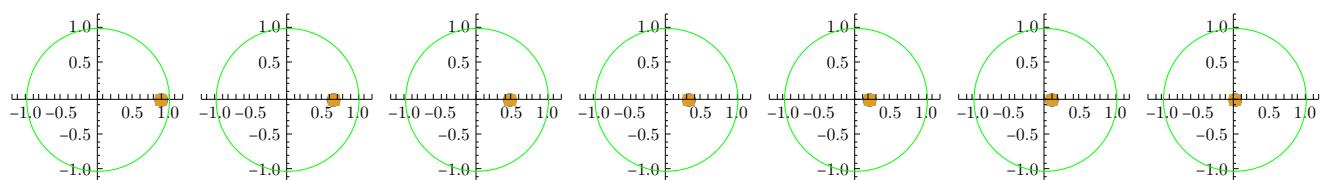


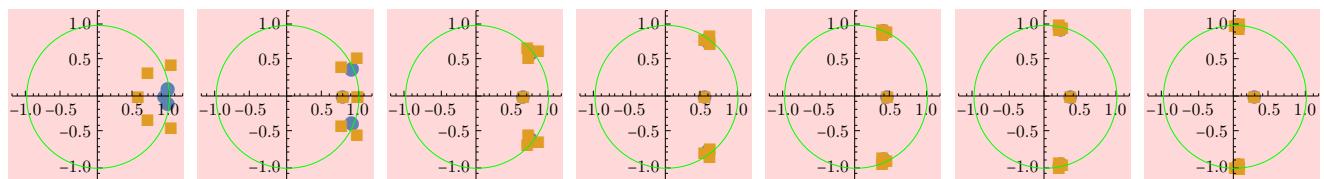
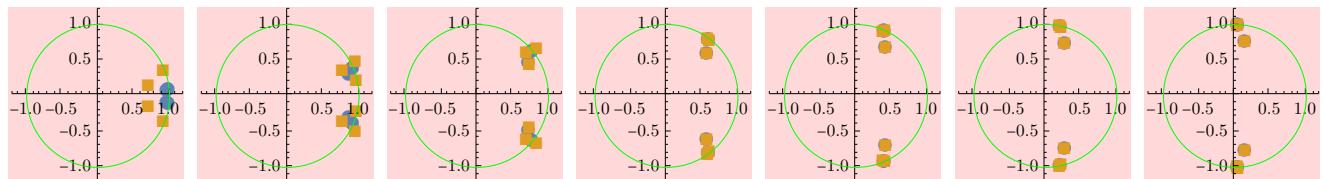
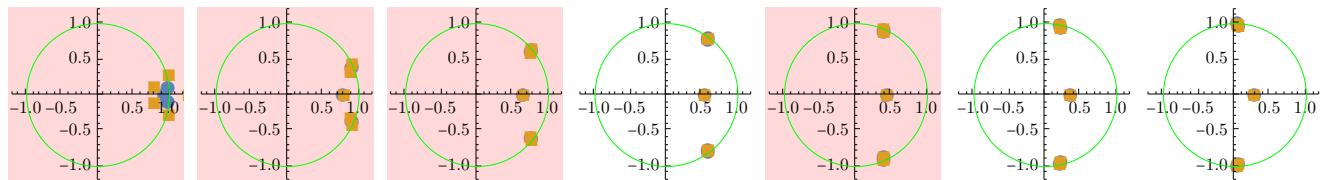
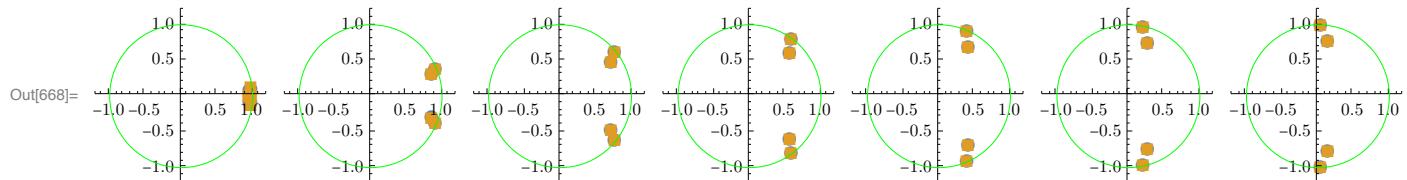




★ Eliptyczne

```
In[668]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





23.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

23.2.1 Dyskretyzacja

```
In[669]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[670]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[671]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

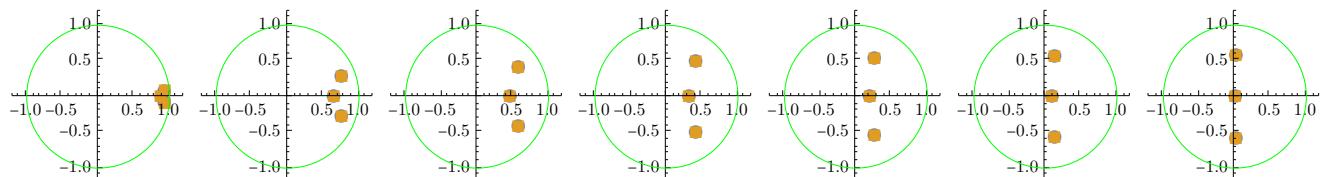
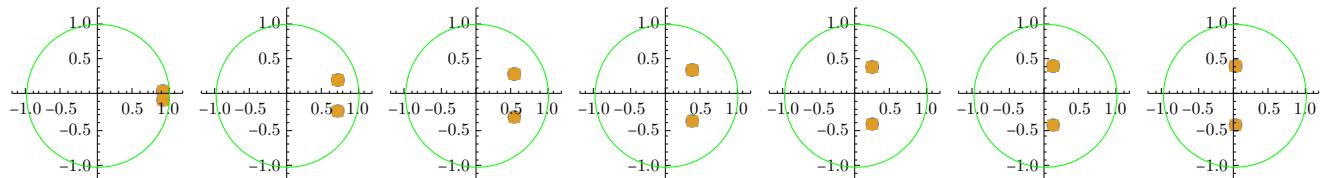
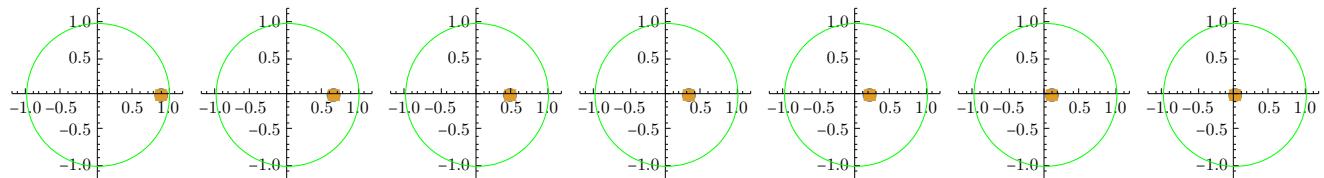
```
In[672]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

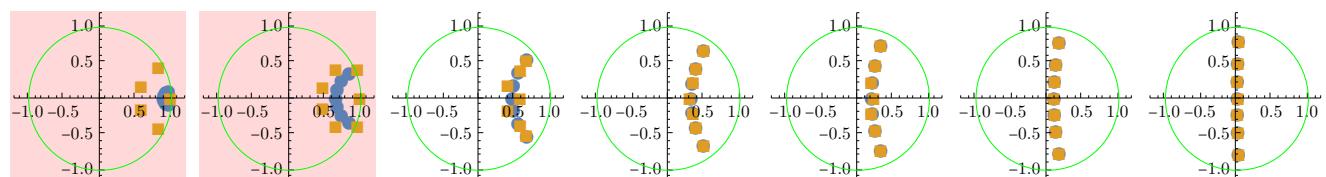
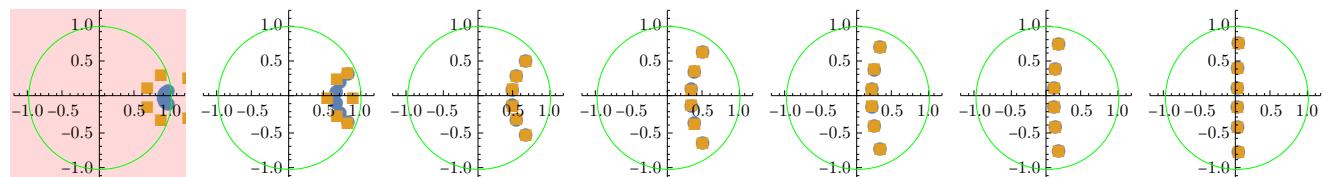
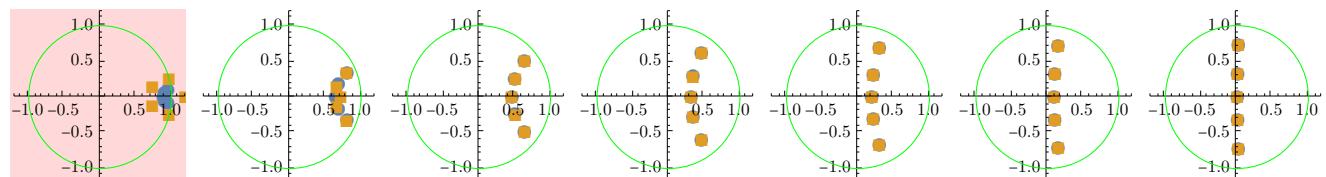
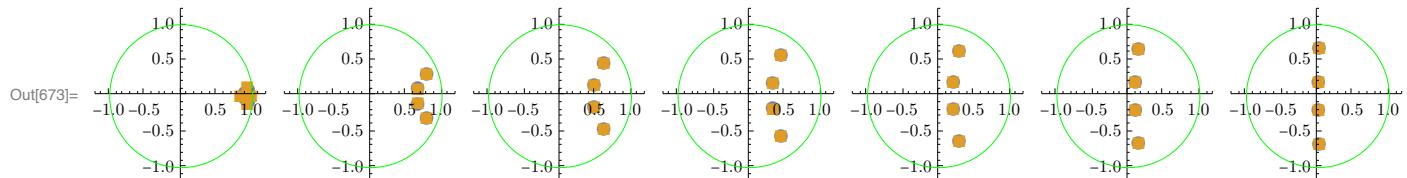
23.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

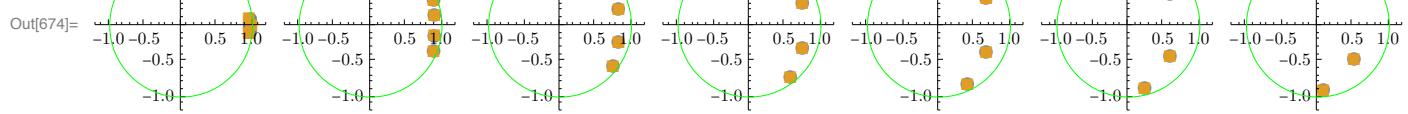
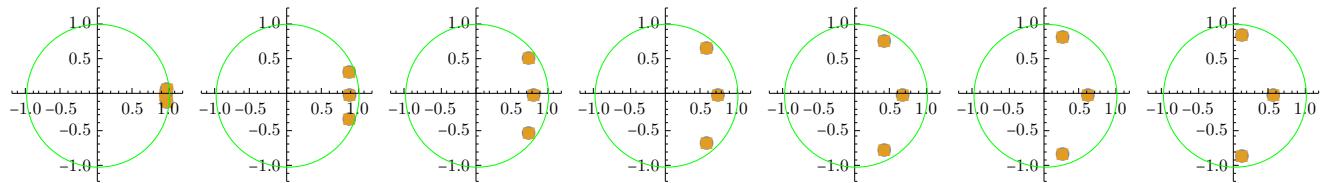
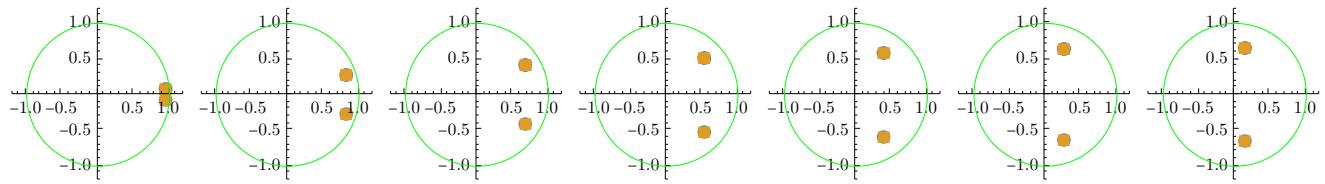
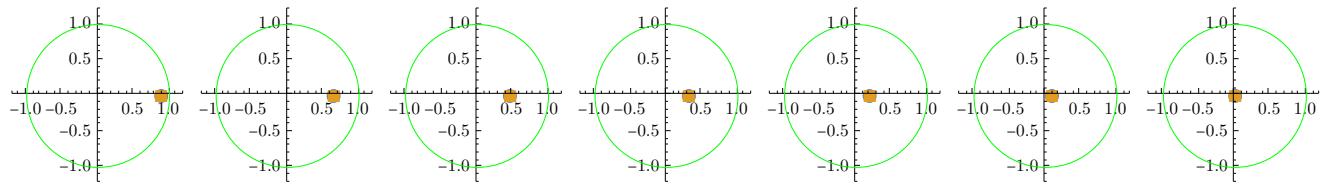
```
In[673]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

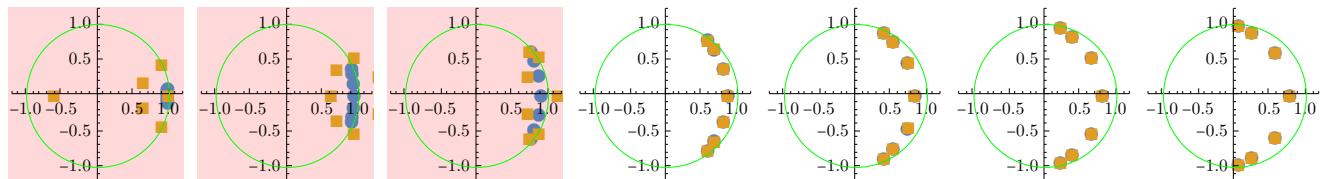
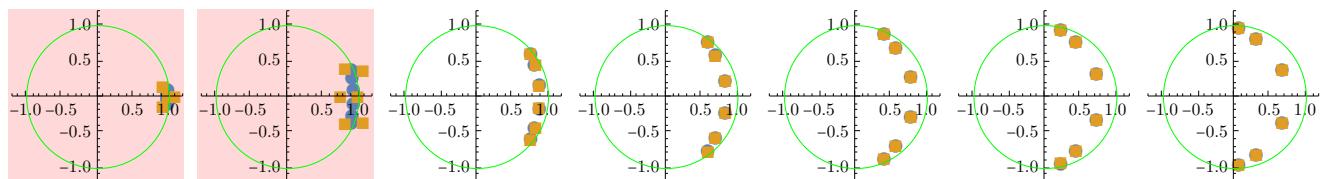
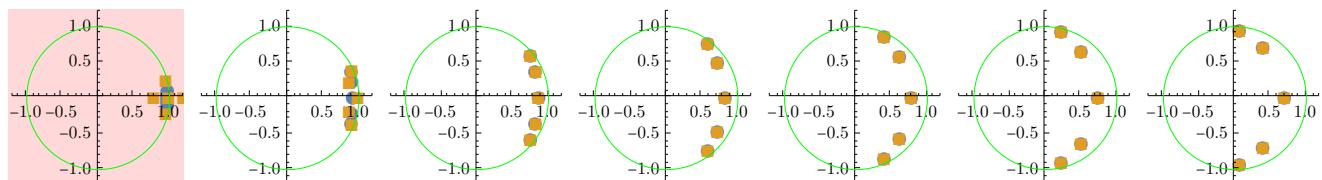




★ Chebyshev 1

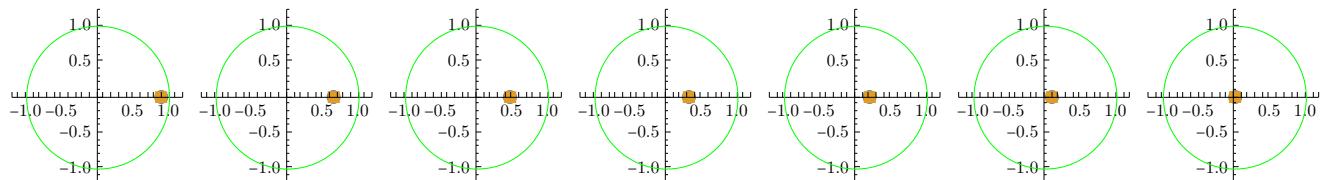
```
In[674]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

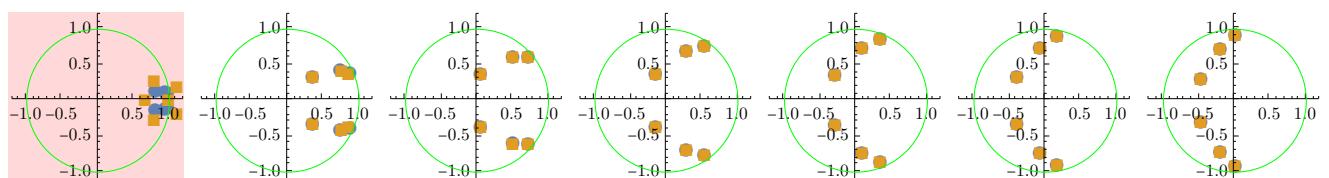
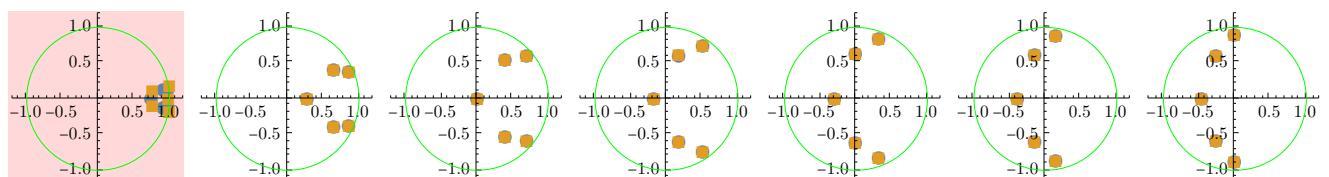
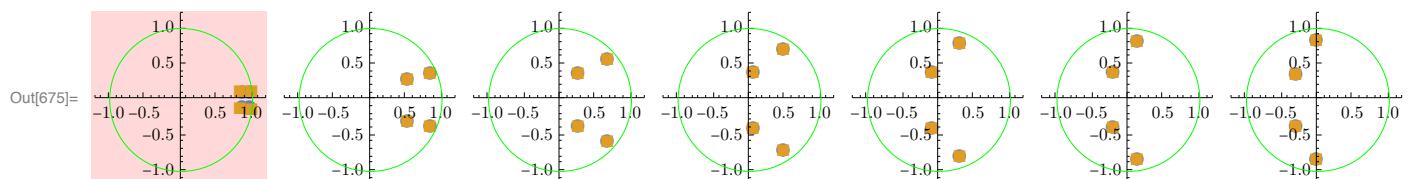
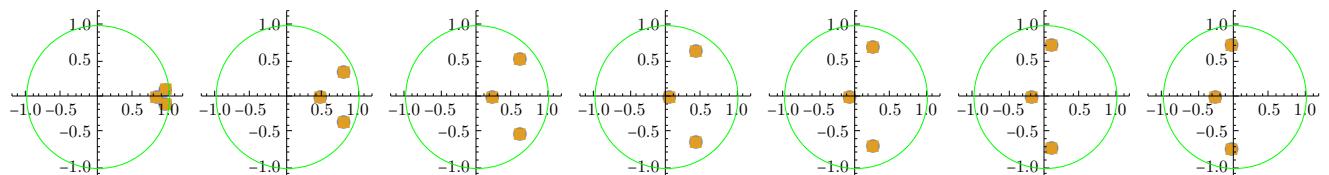
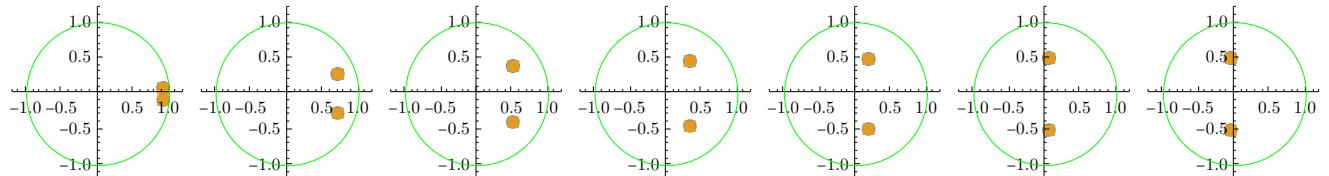


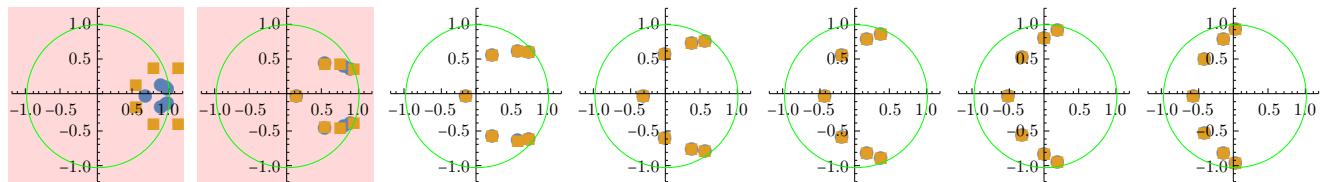


★ Chebyshev 2

```
In[675]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

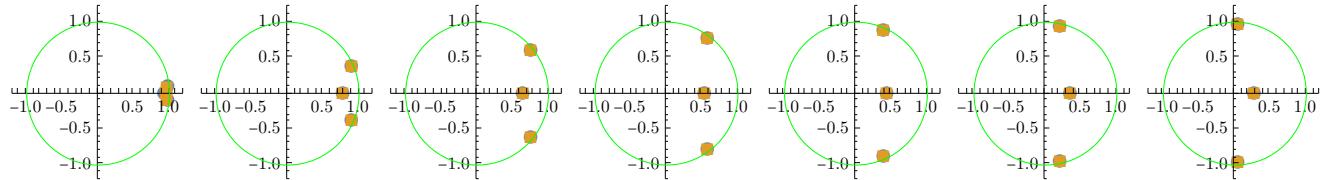
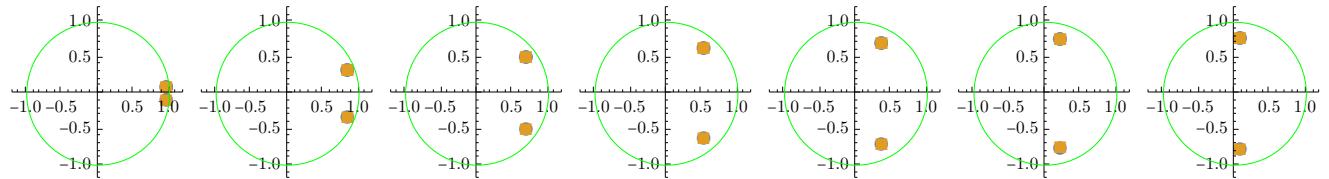
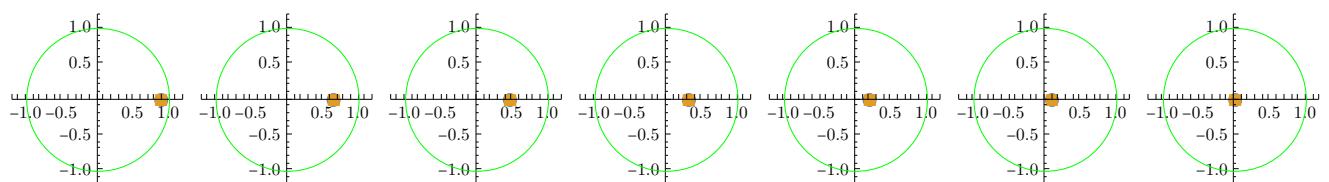


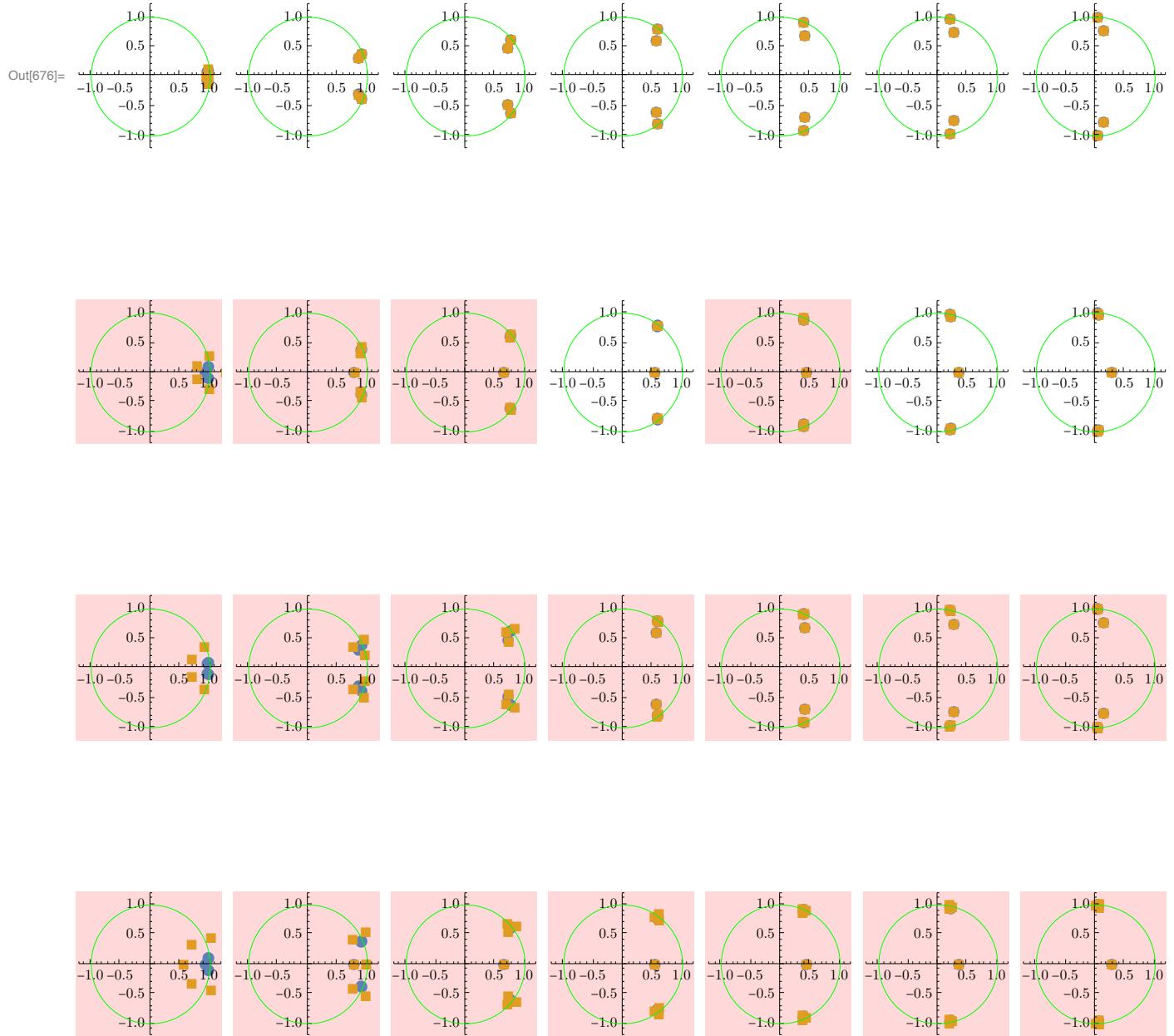




★ Eliptyczne

```
In[676]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





24 Badanie stabilności filtrów cyfrowych (15 bit)

```
In[677]:= bity = 15;
```

24.1 Dyskretyzacja na poziomie współczynników

24.1.1 Dyskretyzacja

```
In[678]:= DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];
```

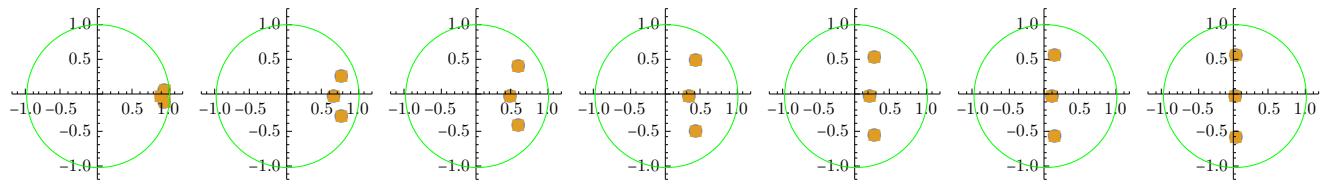
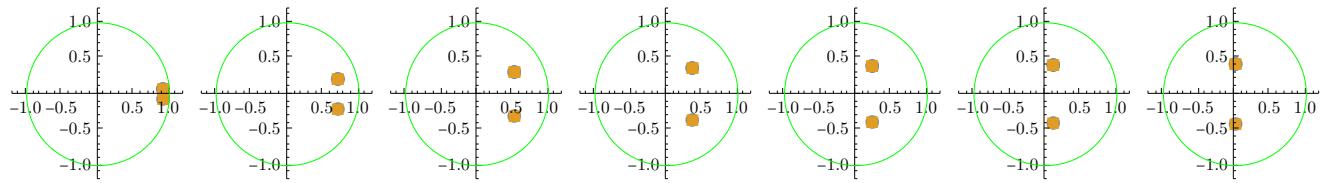
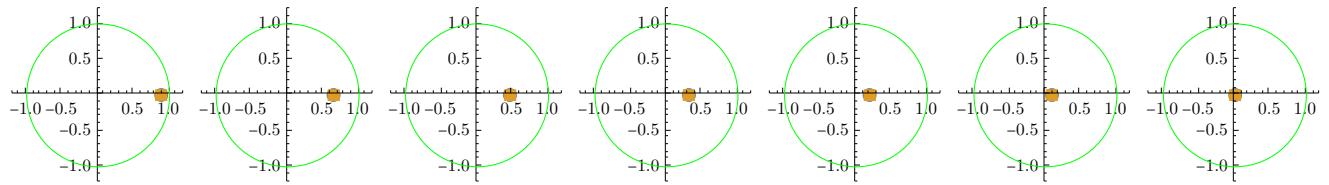
```
In[679]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[680]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[681]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

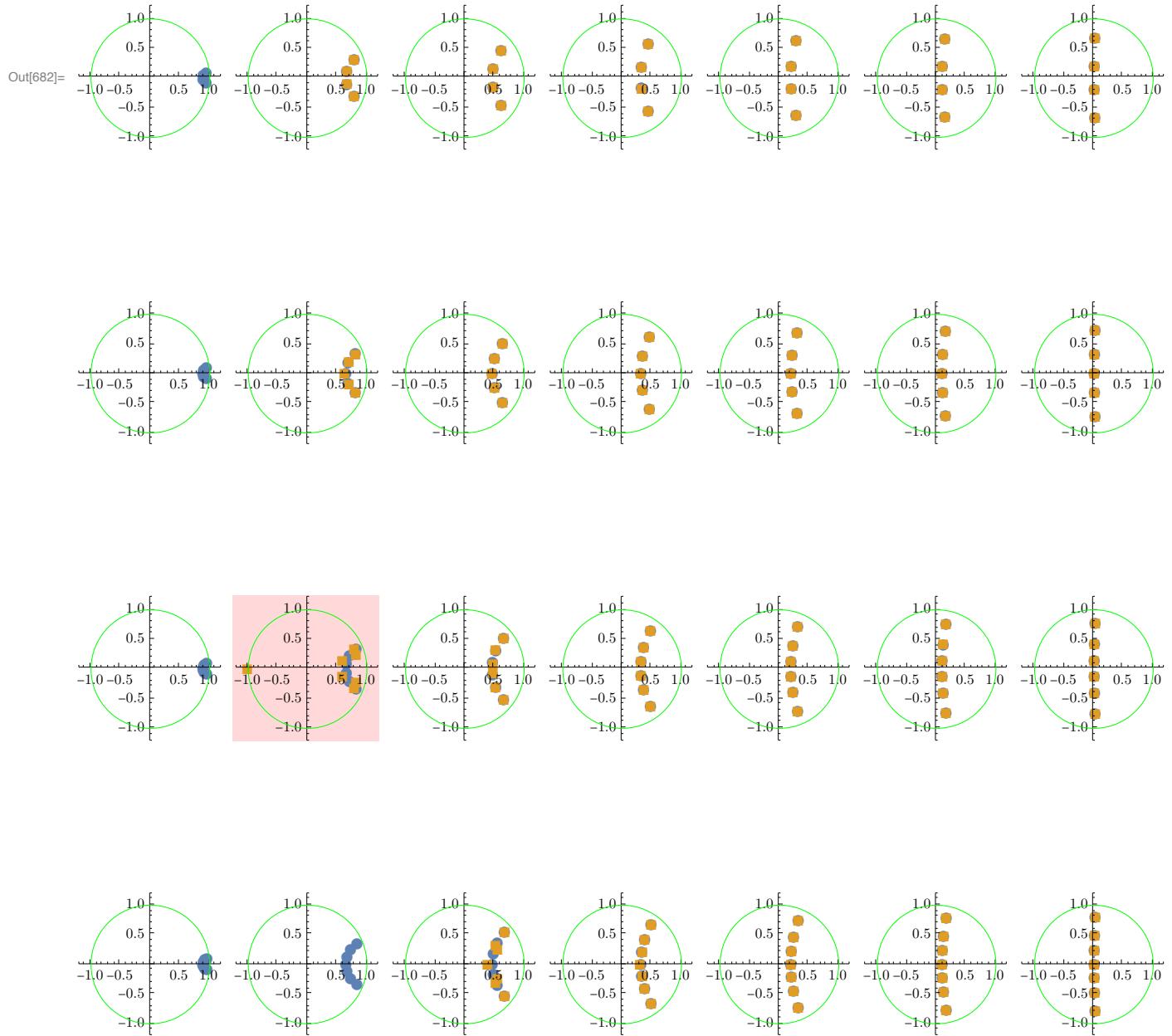
24.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

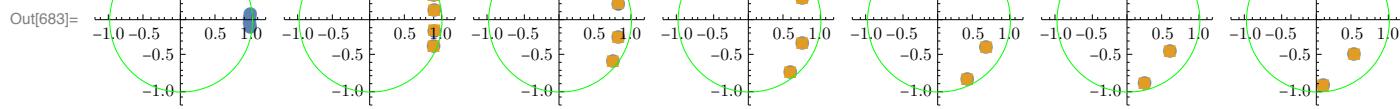
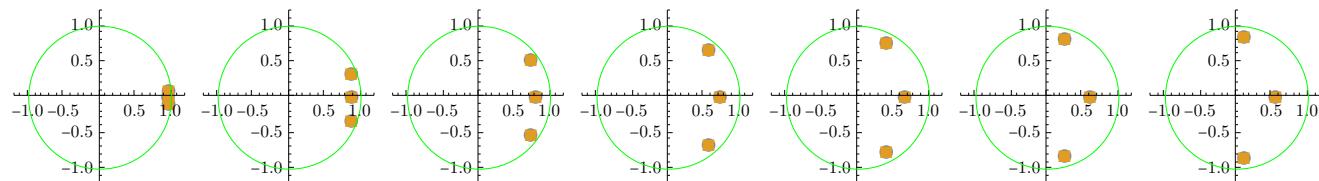
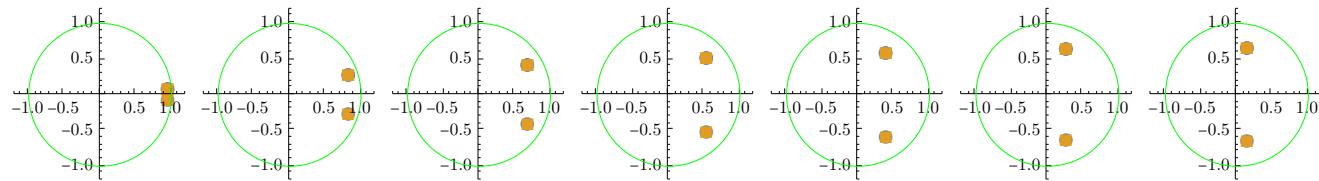
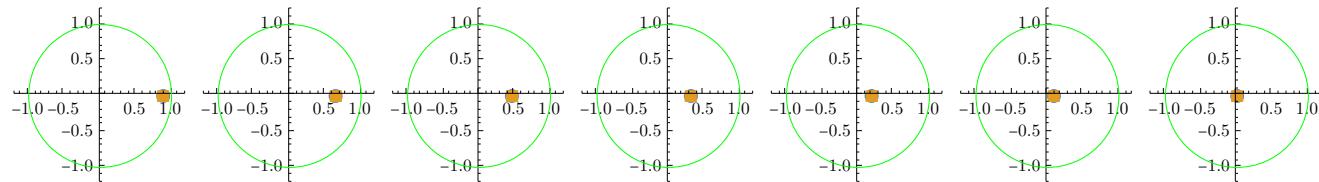
```
In[682]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

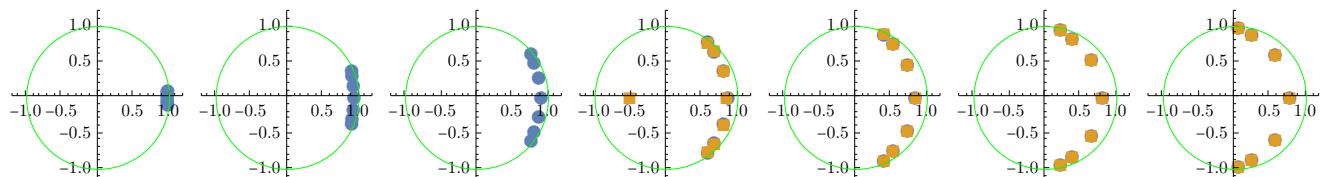
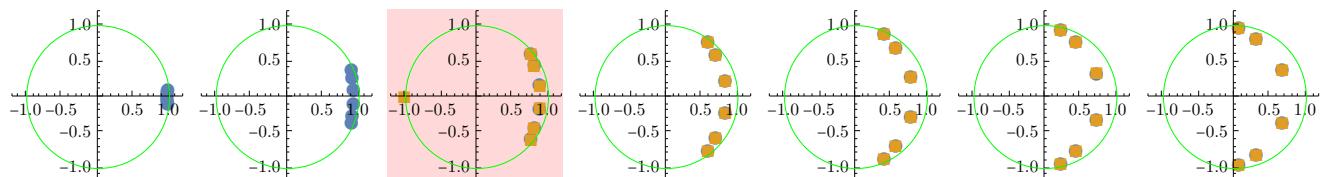
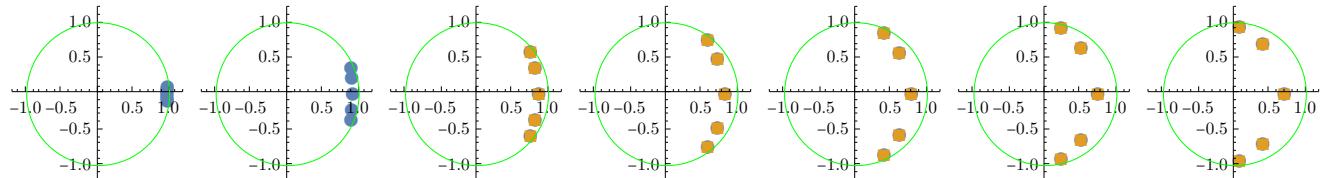




★ Chebyshev 1

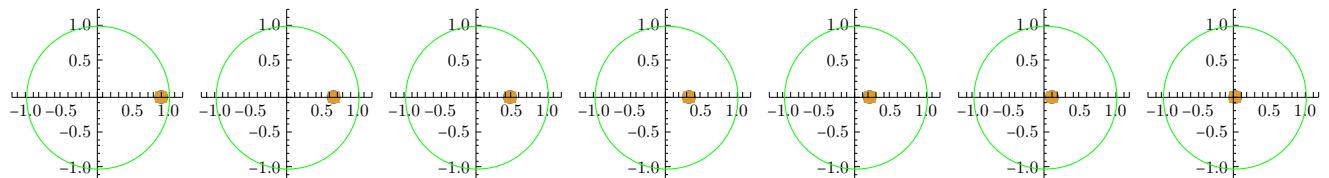
```
In[683]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

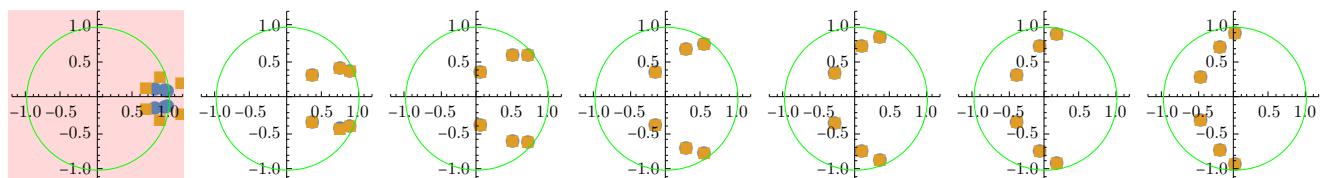
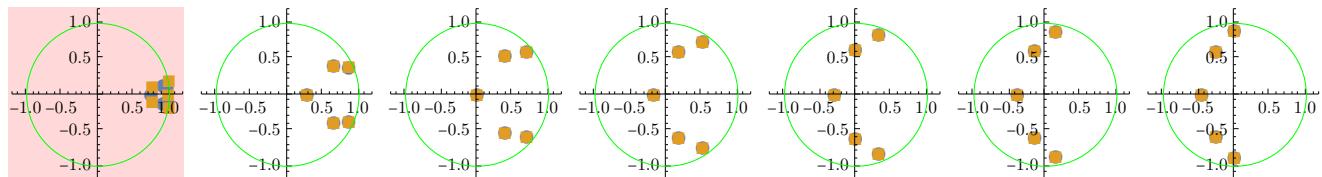
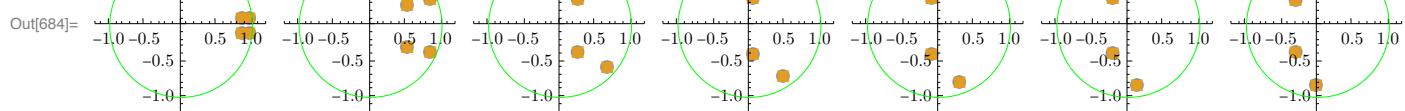
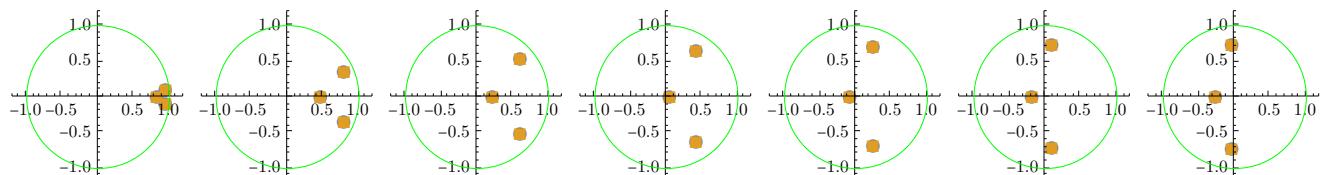
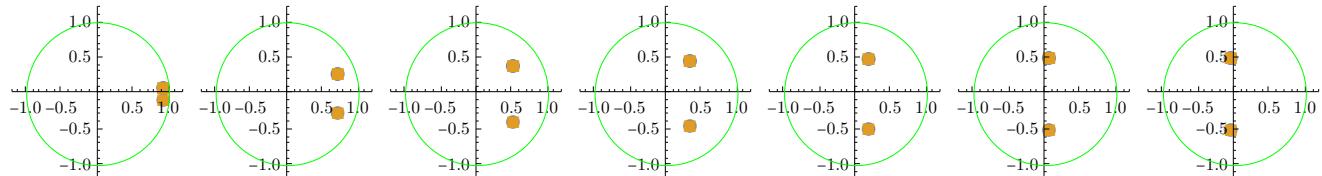


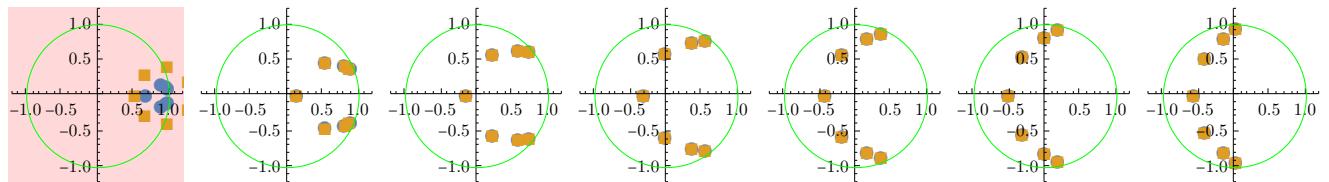


★ Chebyshev 2

```
In[684]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

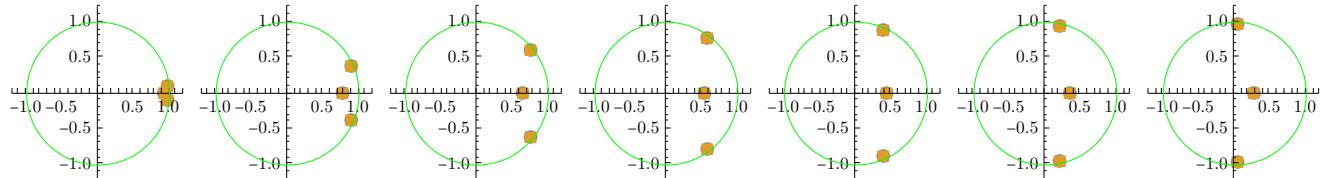
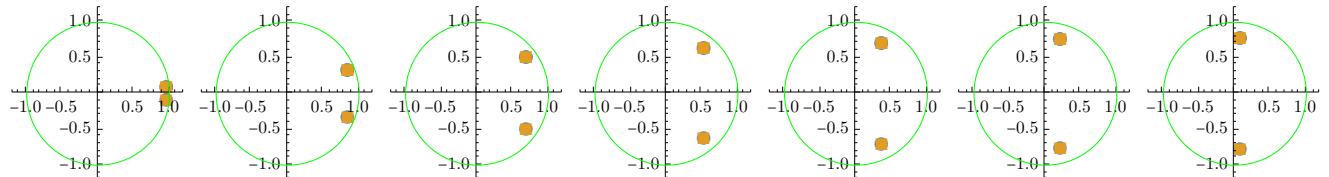
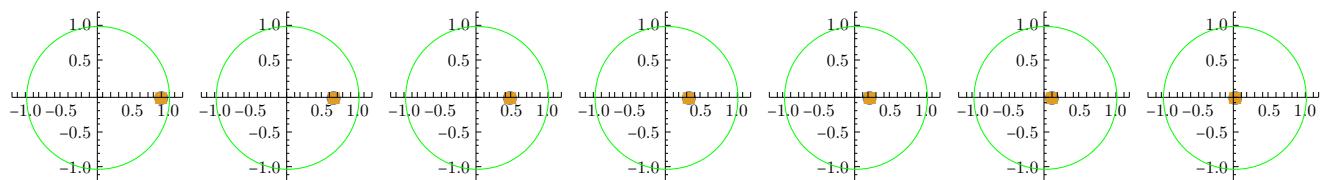


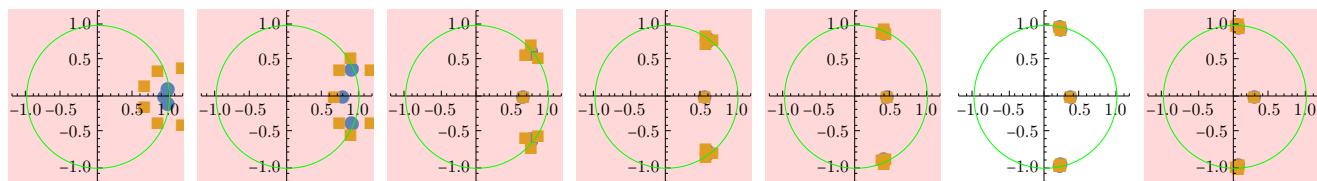
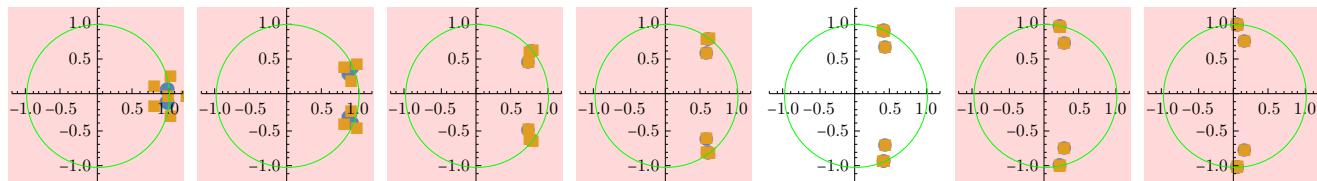
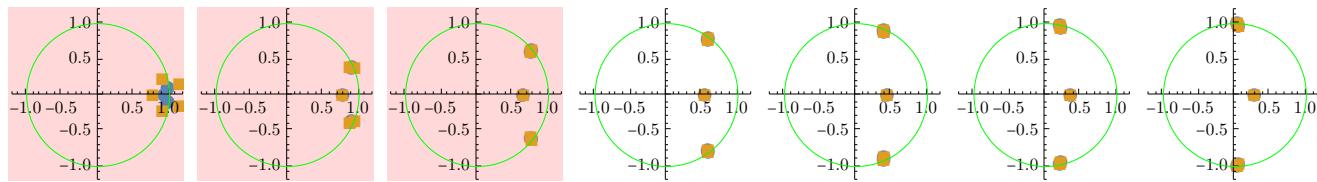
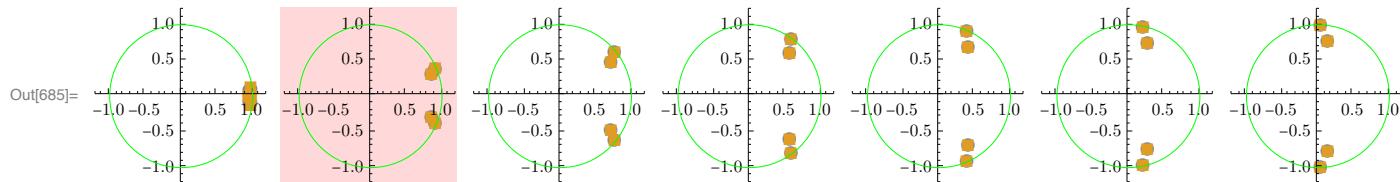




★ Eliptyczne

```
In[685]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





24.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

24.2.1 Dyskretyzacja

```
In[686]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];
```

```
In[687]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[688]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

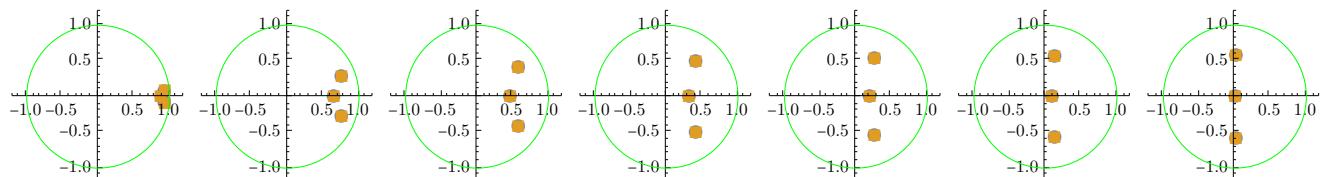
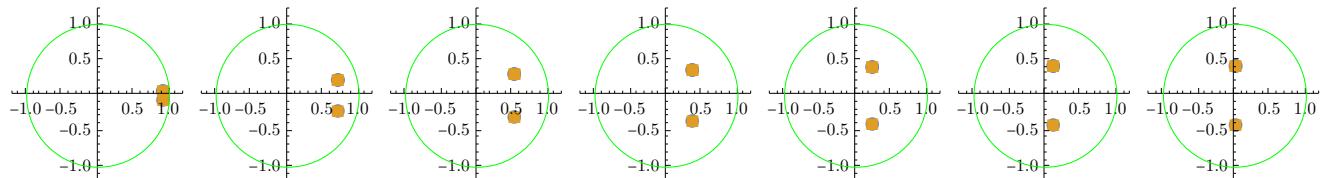
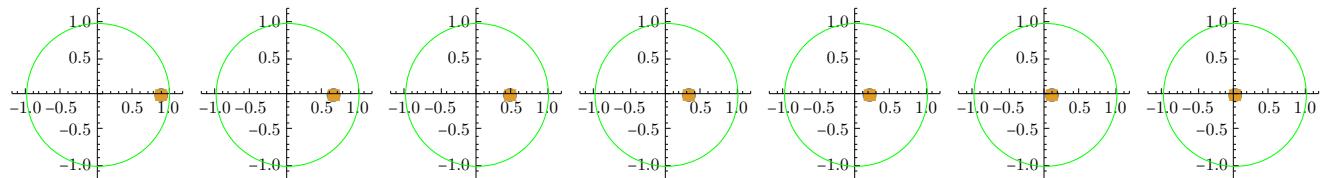
```
In[689]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

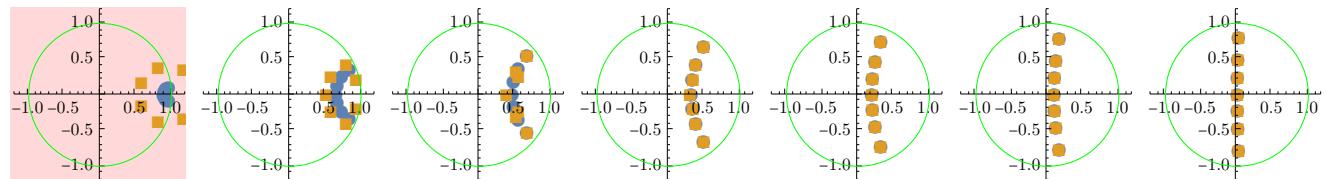
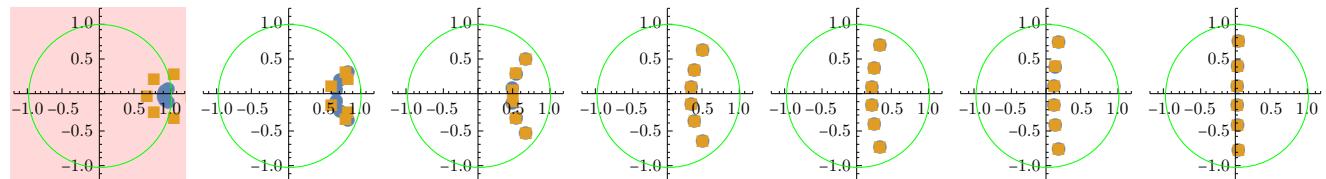
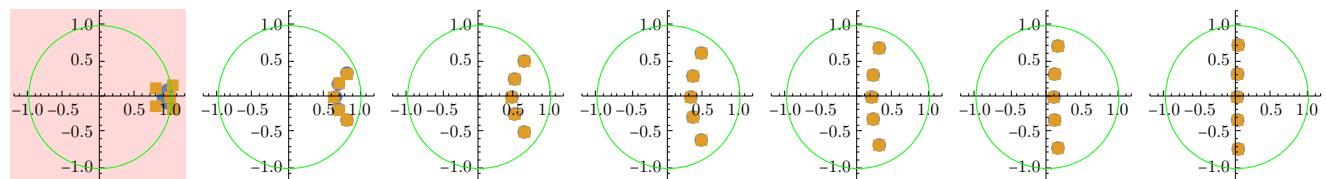
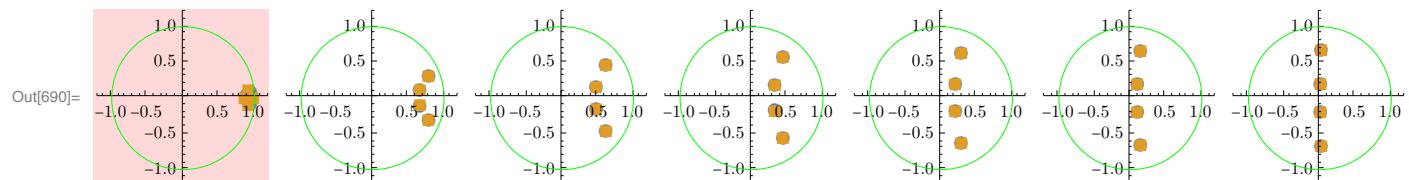
24.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

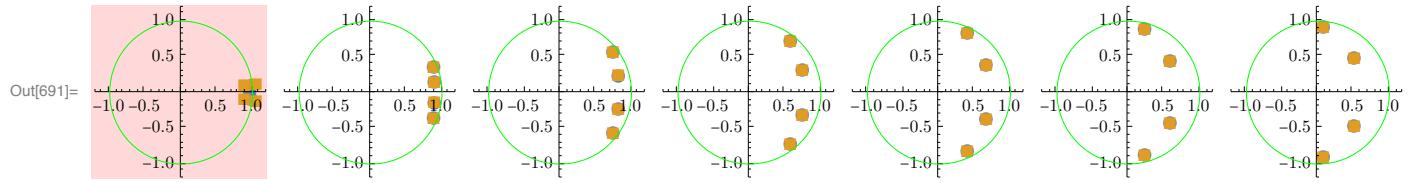
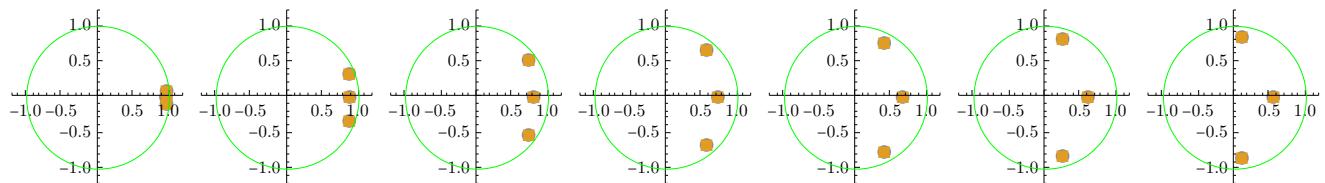
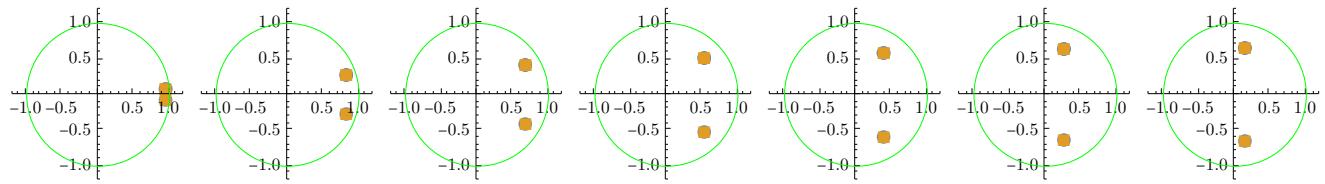
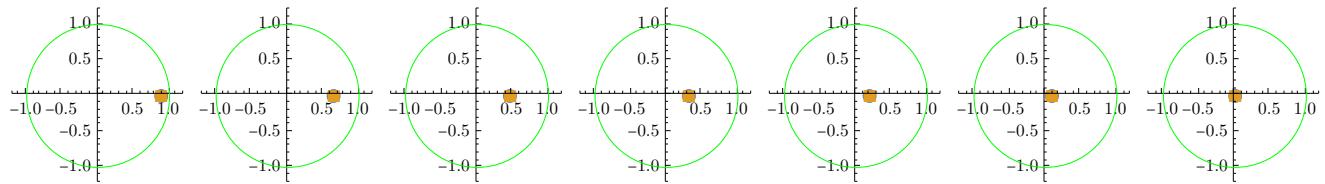
```
In[690]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

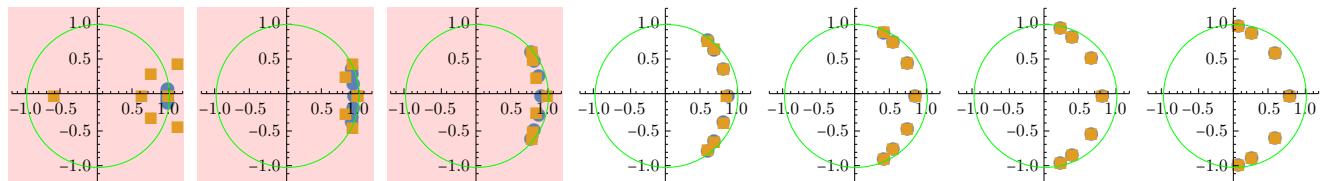
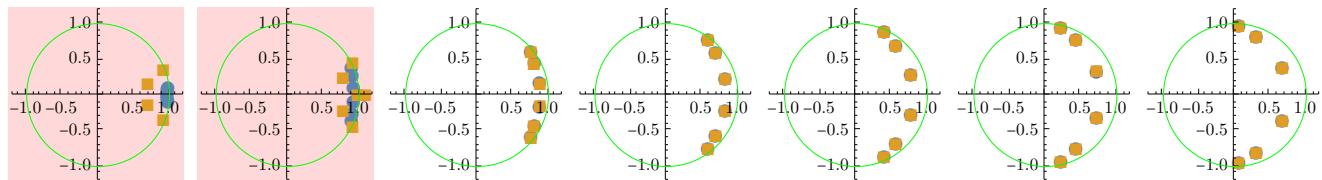
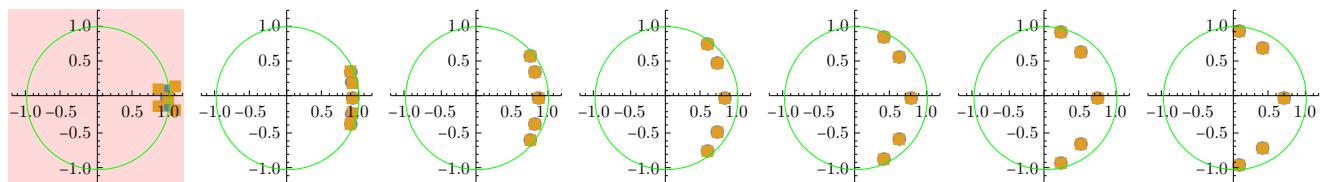




★ Chebyshev 1

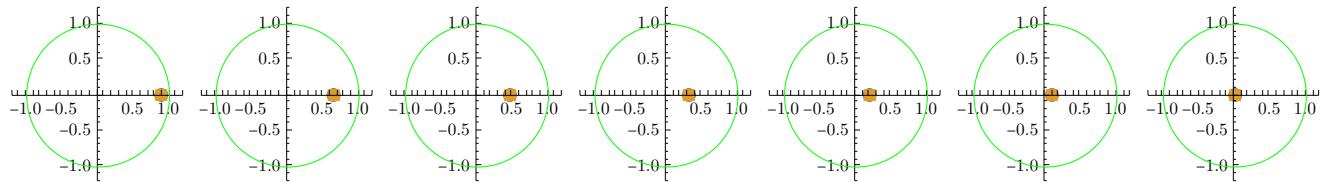
```
In[691]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

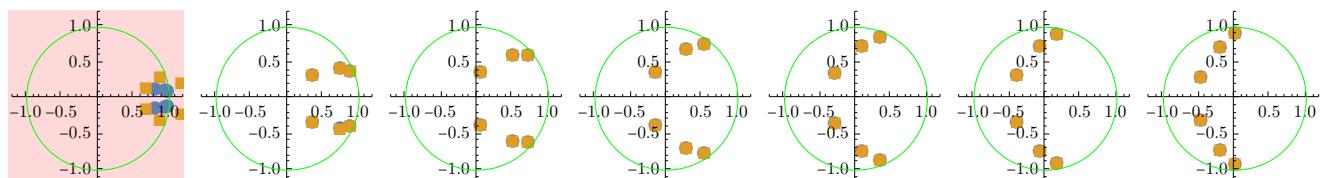
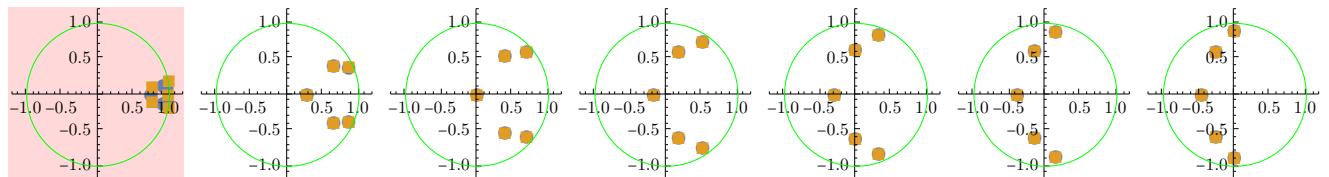
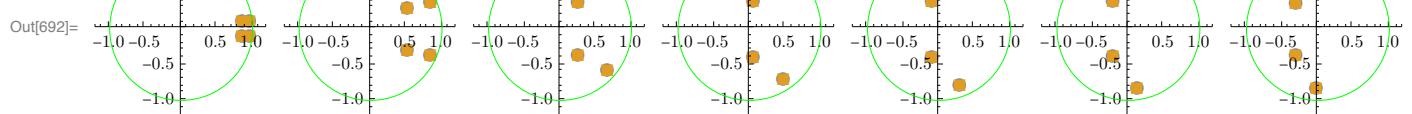
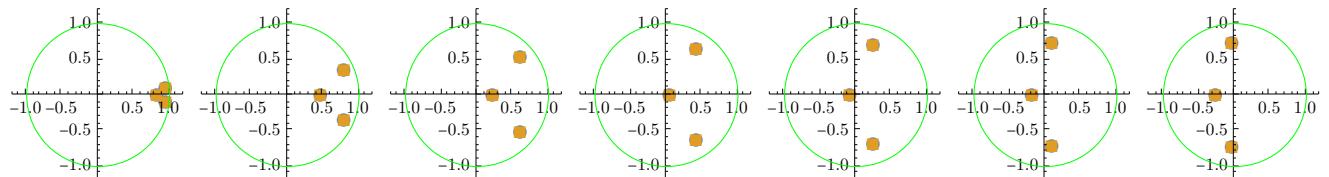
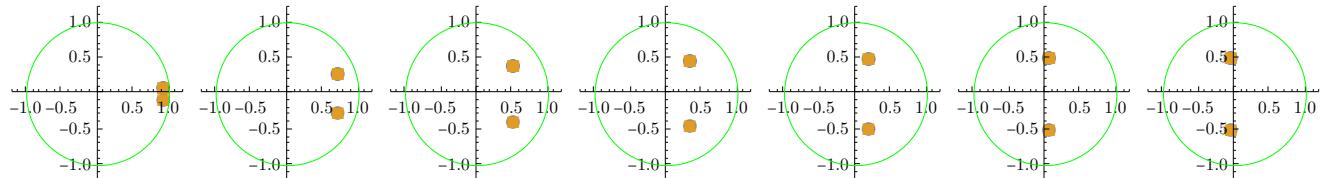


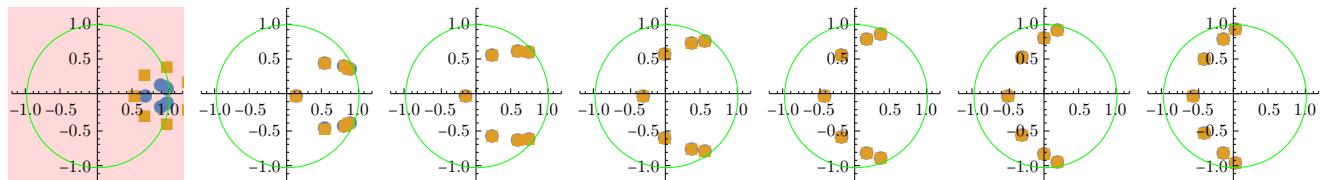


★ Chebyshev 2

```
In[692]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

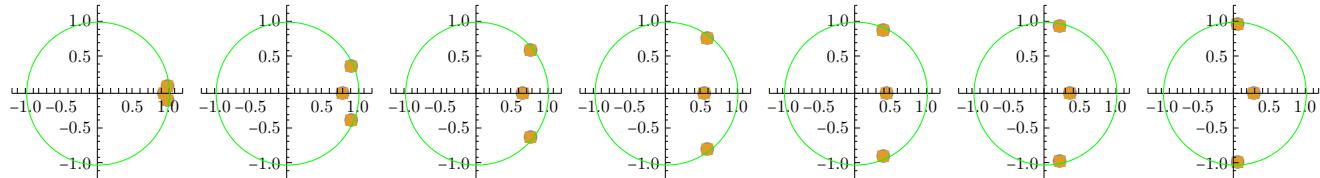
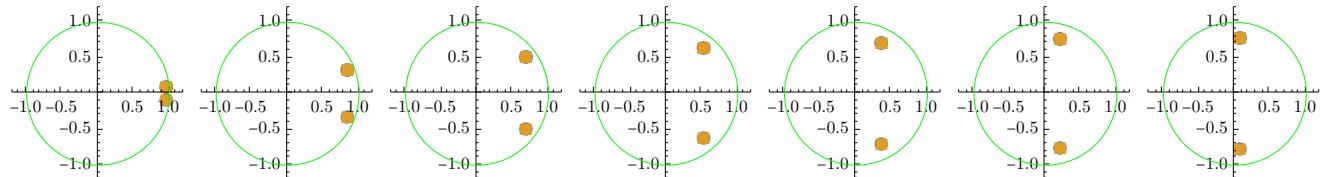
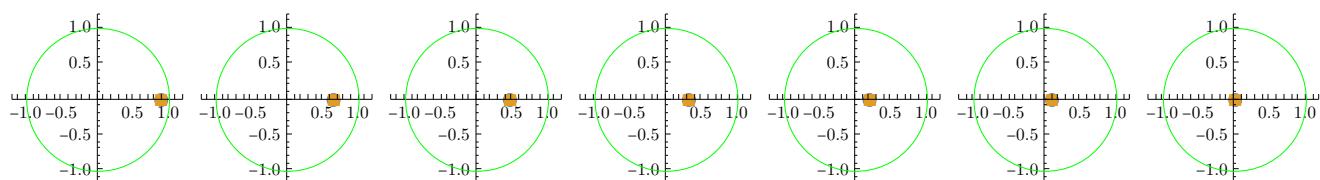


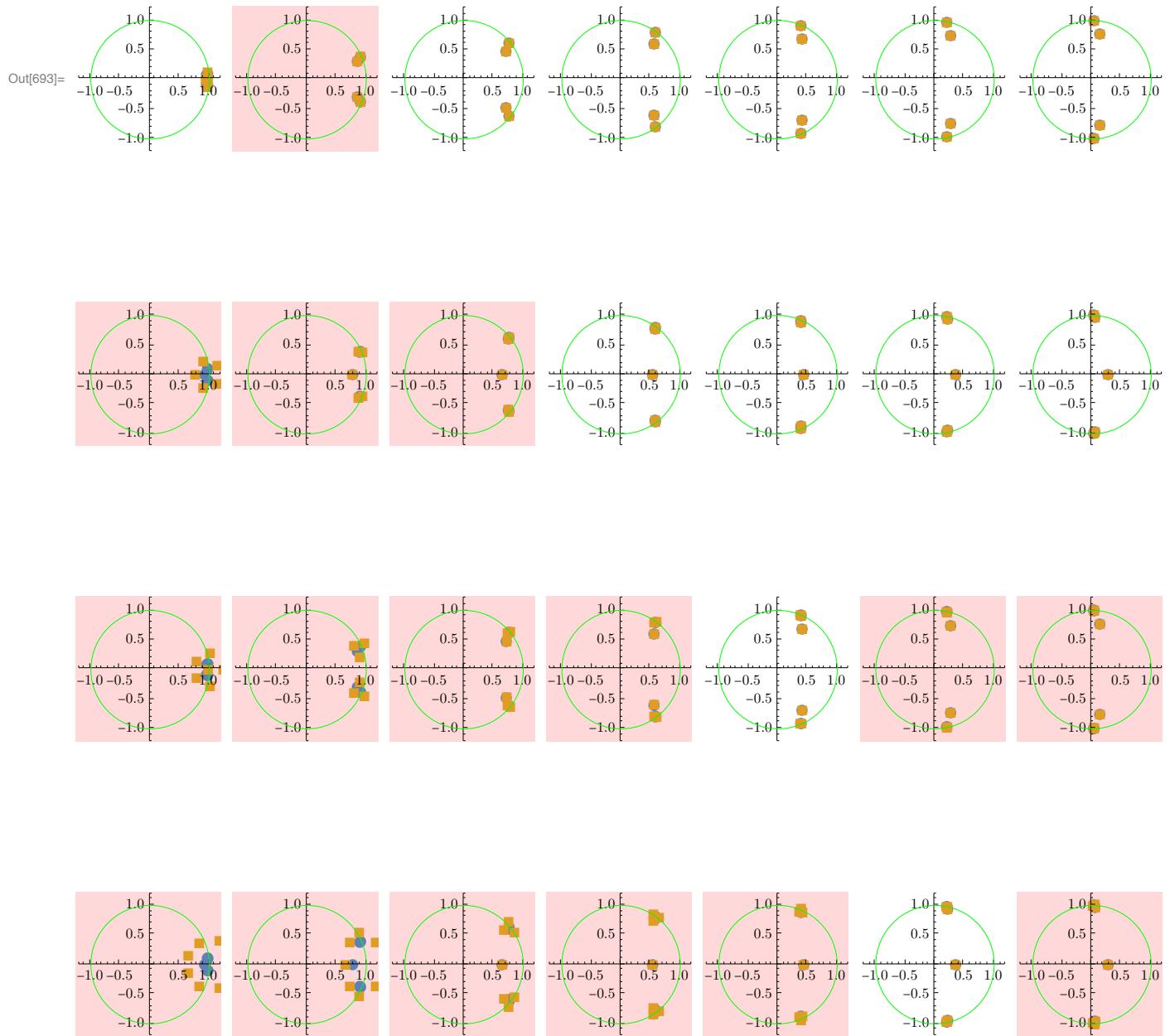




★ Eliptyczne

```
In[693]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





25 Badanie stabilności filtrów cyfrowych (16 bit)

In[694]:= **bity = 16;**

25.1 Dyskretyzacja na poziomie współczynników

25.1.1 Dyskretyzacja

In[695]:= **DGbmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGbmodels, {2}];**

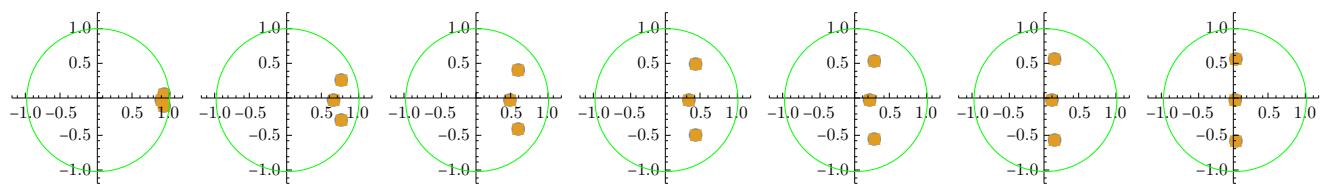
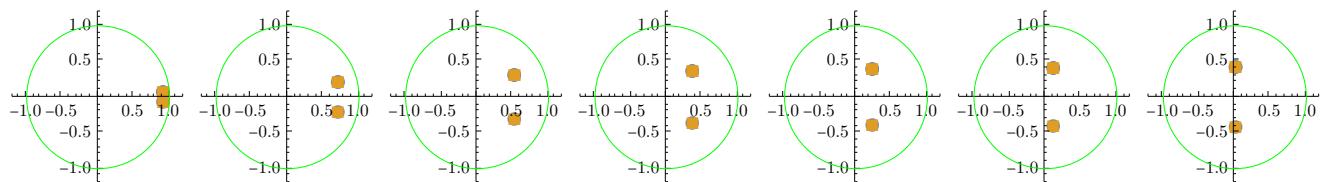
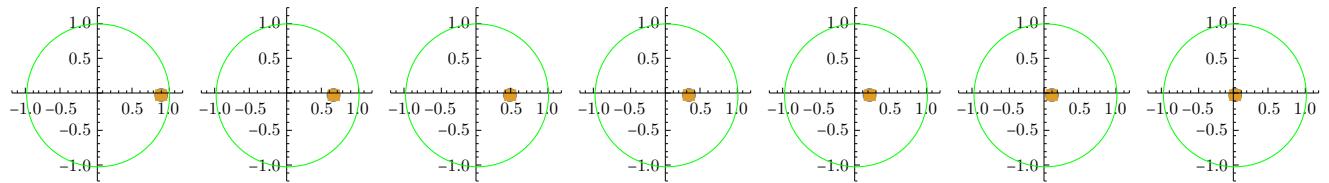
```
In[696]:= DGc1modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc1models, {2}];  
In[697]:= DGc2modelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGc2models, {2}];  
In[698]:= DGmodelsDc = Map[DiscretizeModelCoeffs[#, bity] &, DGmodels, {2}];
```

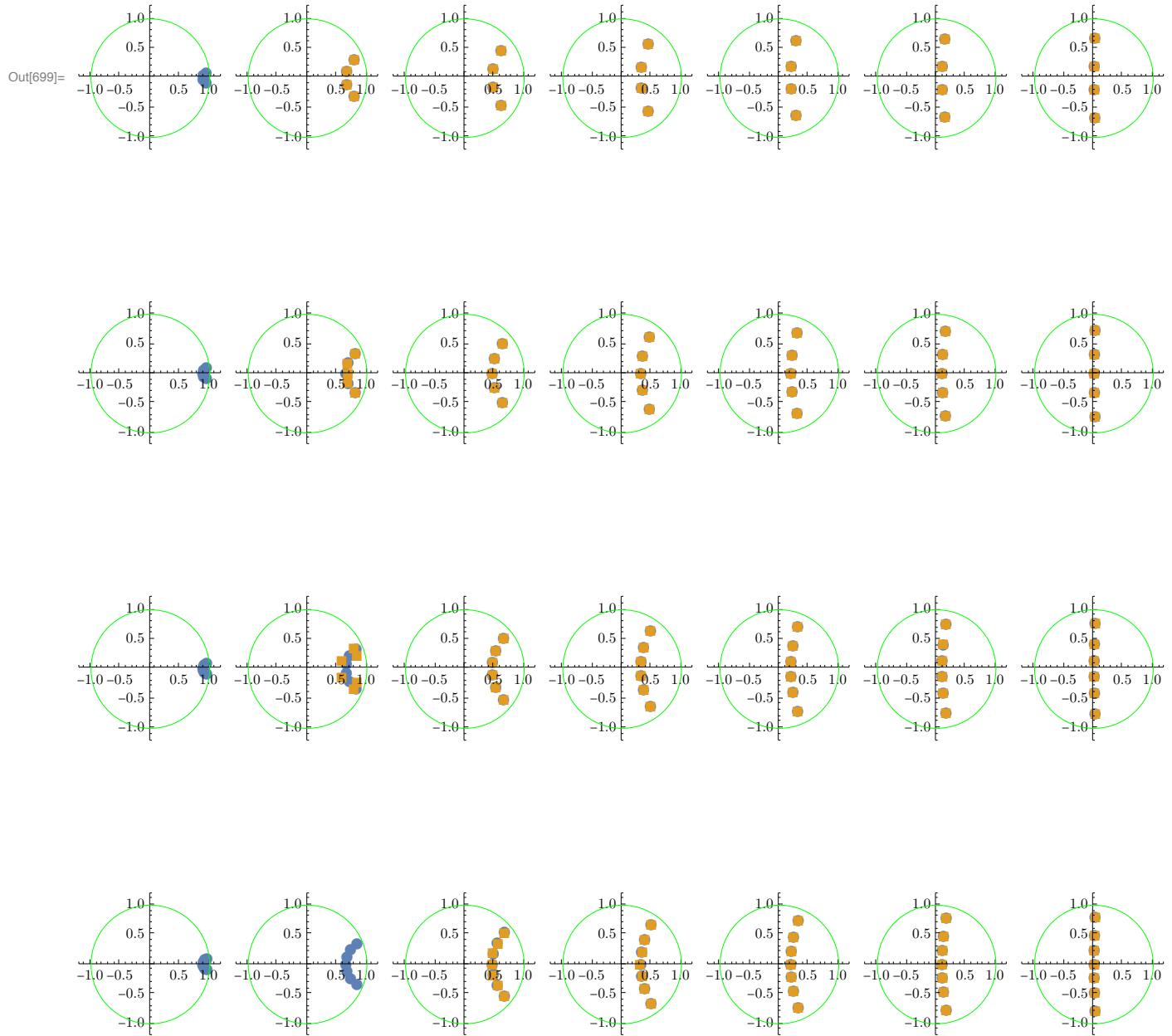
25.1.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rzząd filtra.

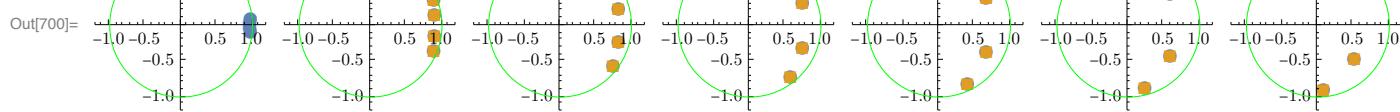
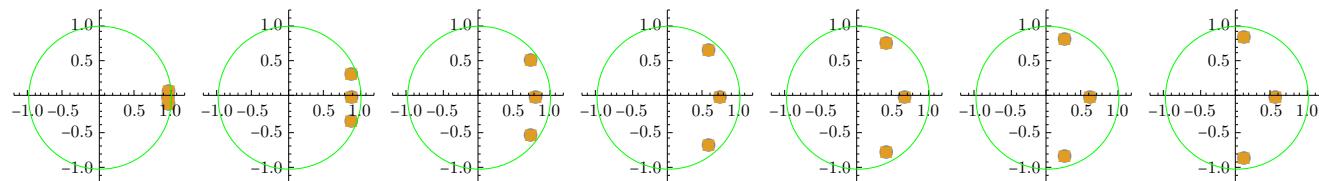
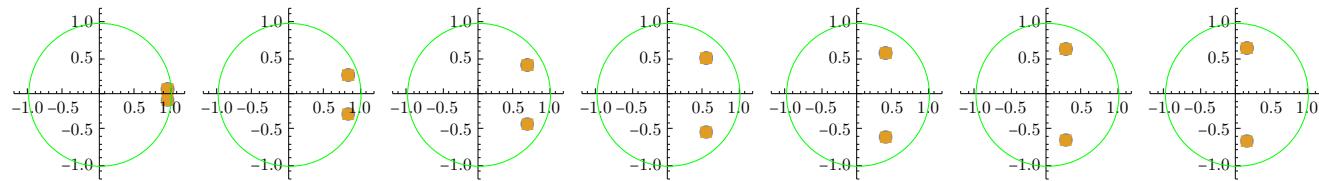
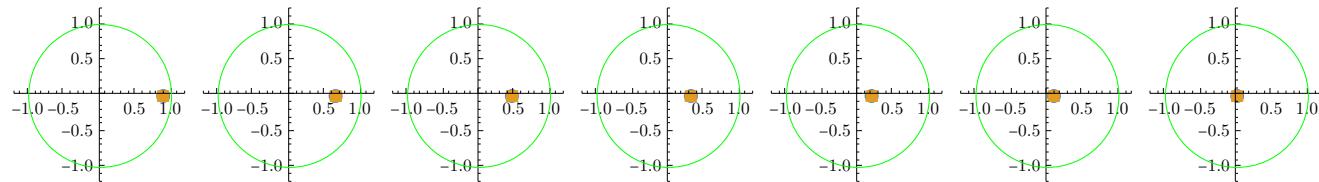
```
In[699]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,  
{DGbmodels, DGbmodelsDc}, 2]
```

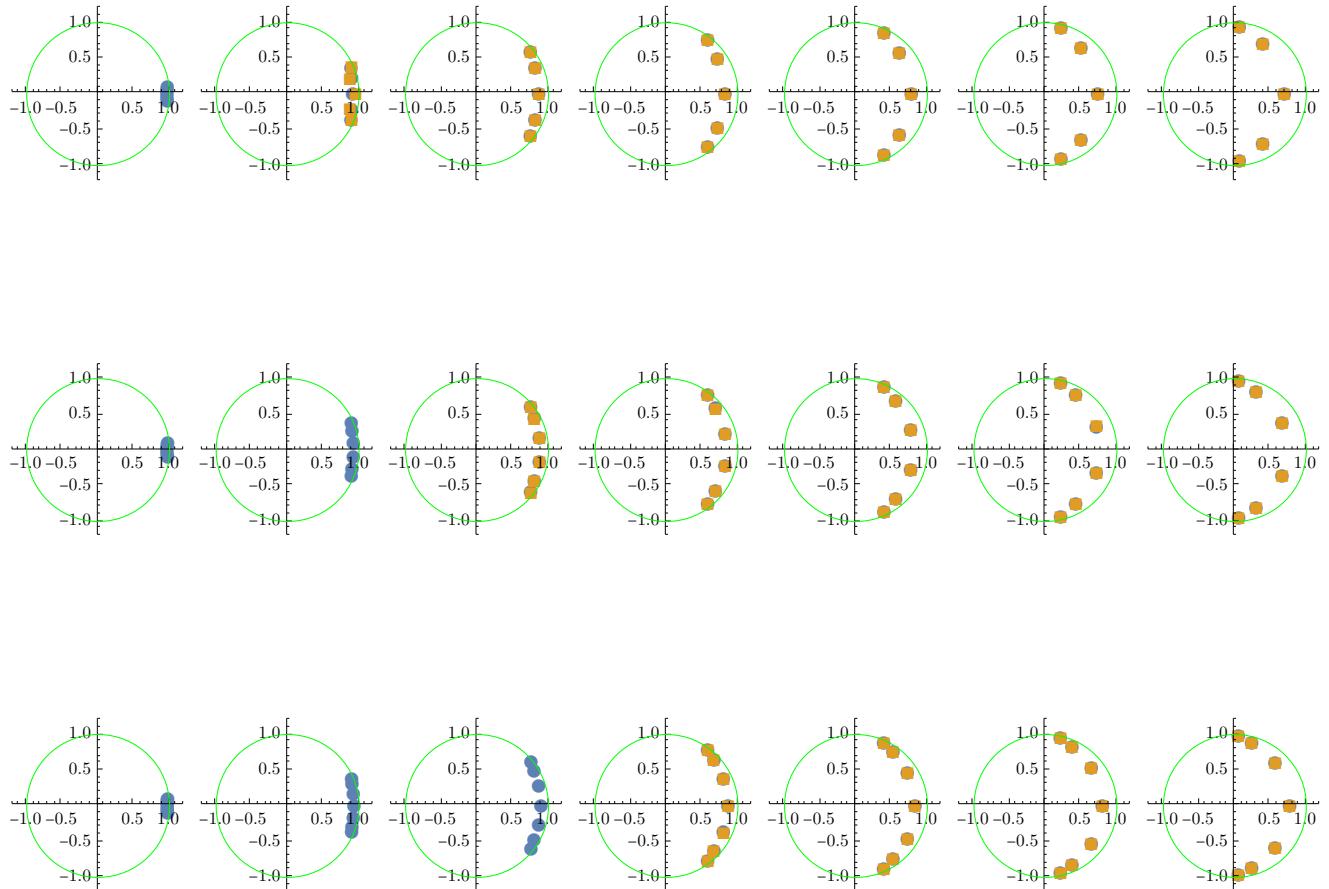




★ Chebyshev 1

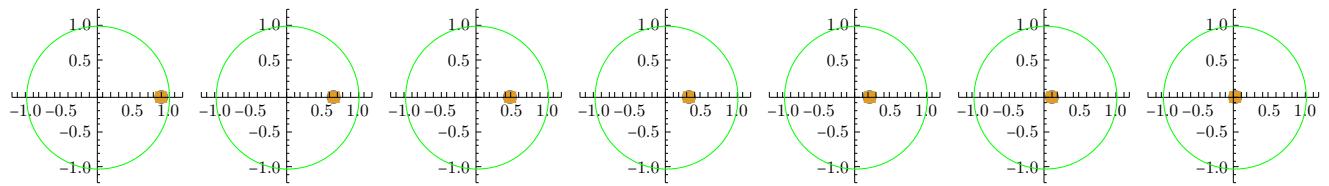
```
In[700]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc}, 2]
```

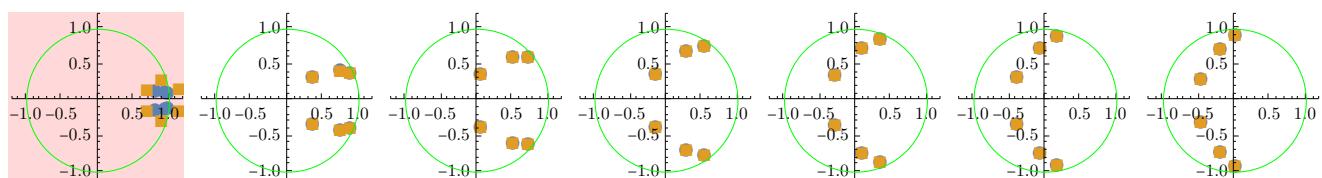
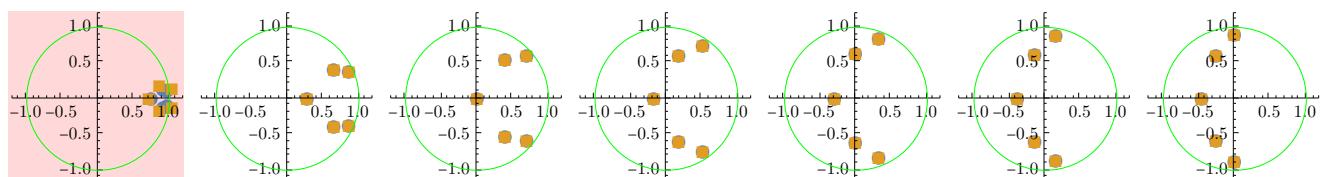
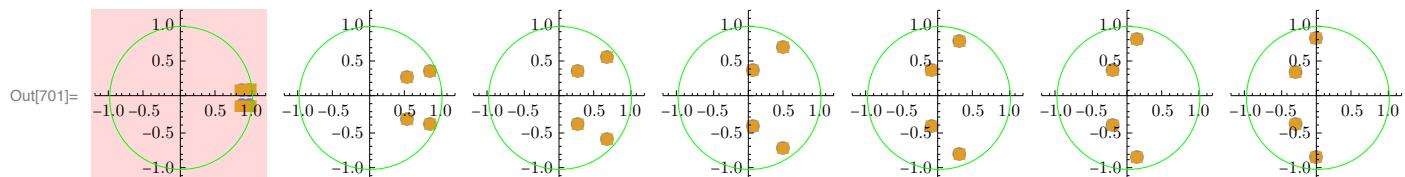
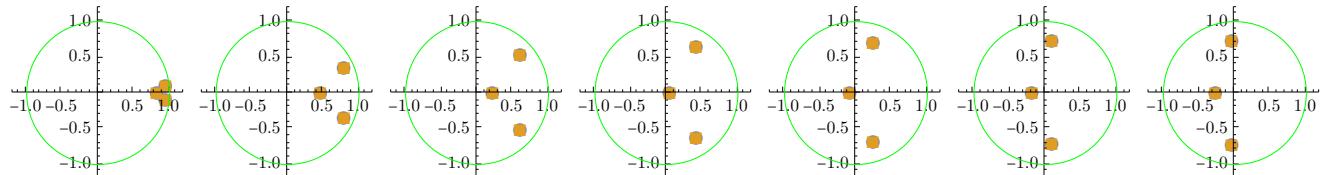
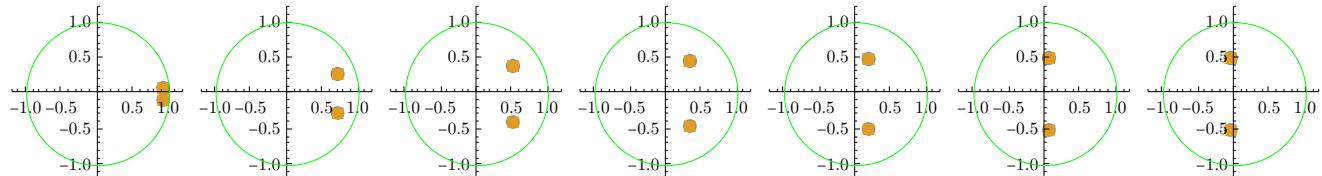


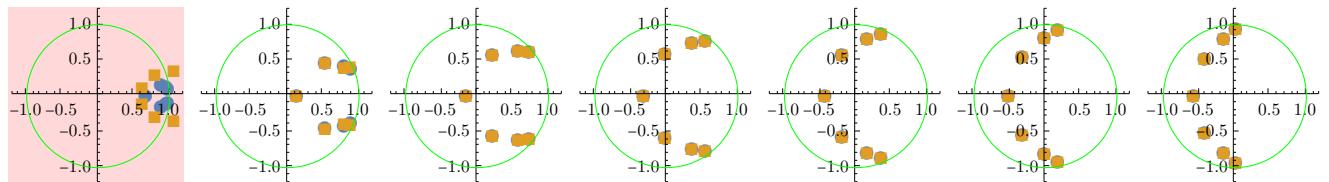


★ Chebyshev 2

```
In[701]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc}, 2]
```

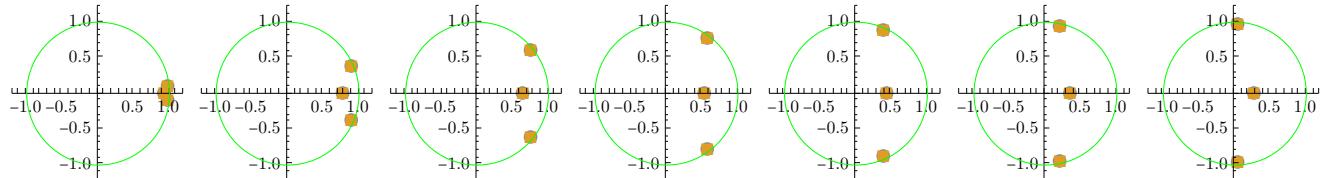
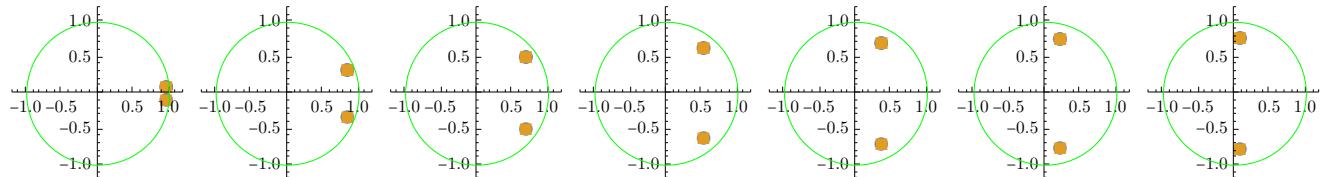
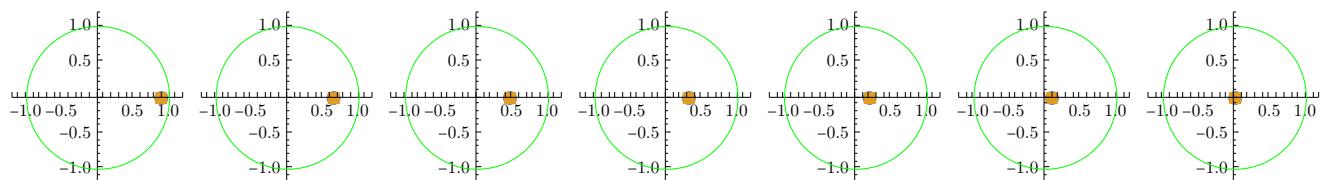


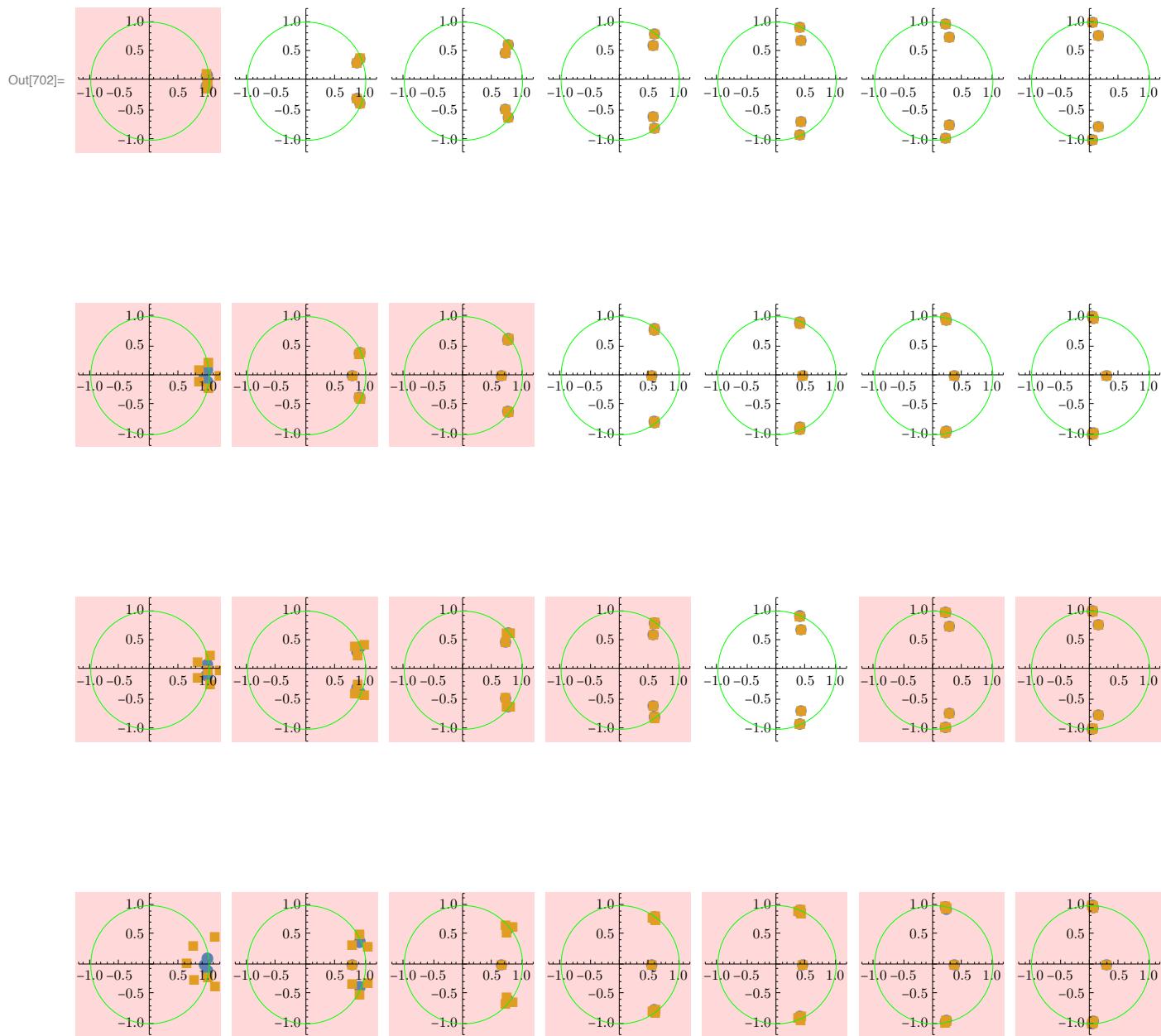




★ Eliptyczne

```
In[702]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc}, 2]
```





25.2 Dyskretyzacja na poziomie współczynników zer i biegunów z osobna

25.2.1 Dyskretyzacja

```
In[703]:= DGbmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGbmodels, {2}];

In[704]:= DGc1modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc1models, {2}];
```

```
In[705]:= DGc2modelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGc2models, {2}];
```

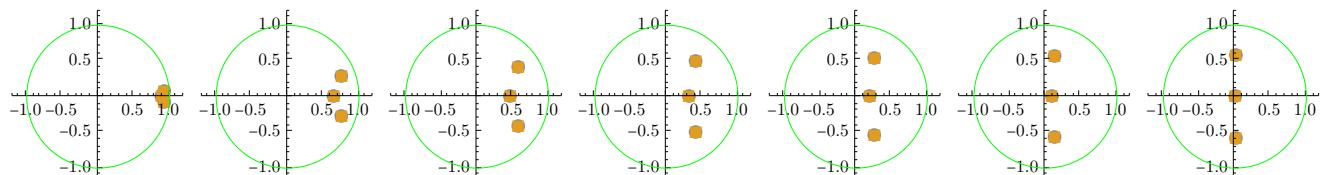
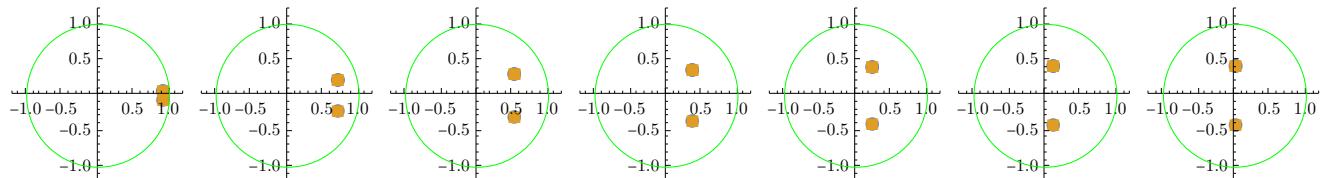
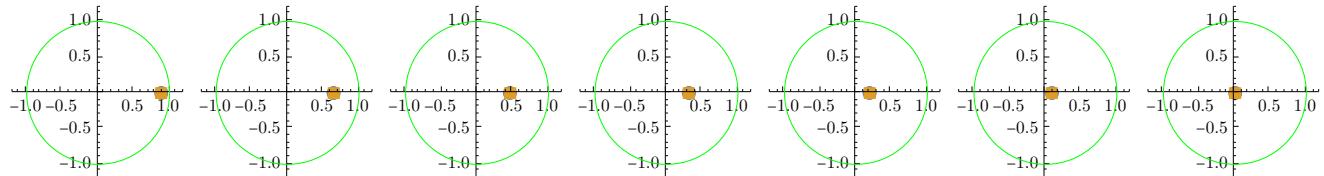
```
In[706]:= DGmodelsDc2 = Map[DiscretizeModelCoeffs2[#, bity] &, DGmodels, {2}];
```

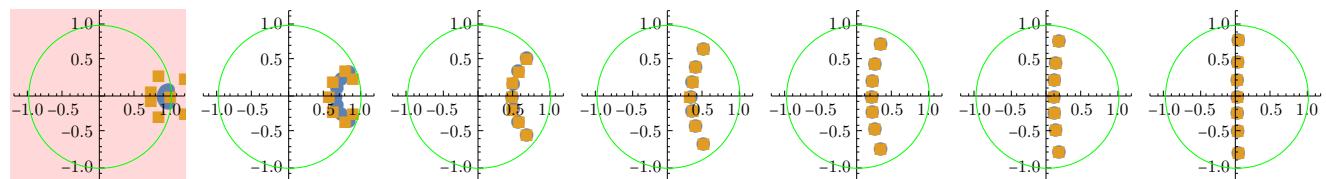
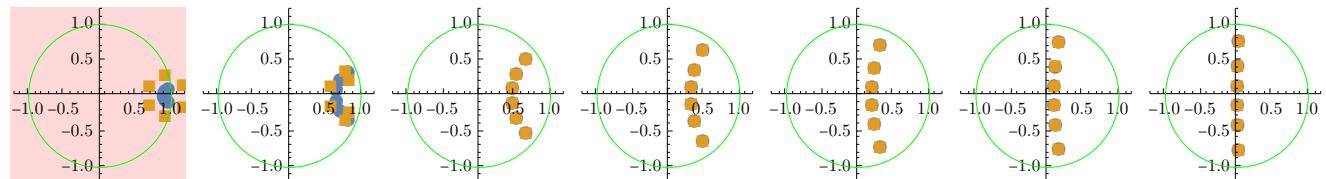
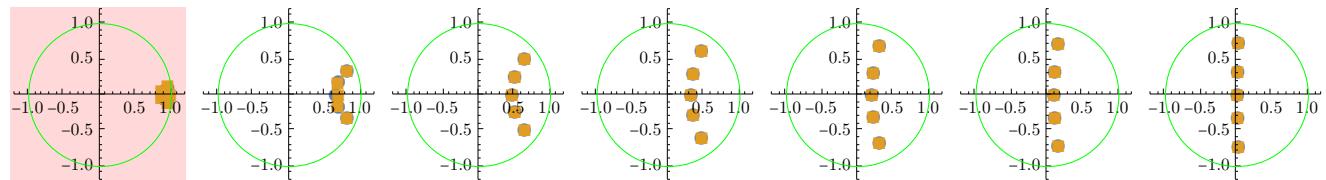
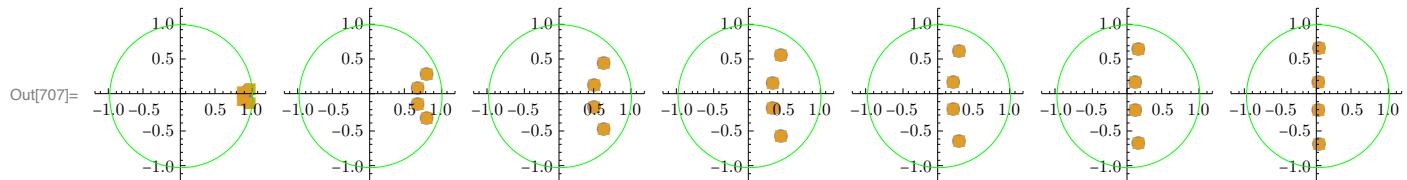
25.2.2 Porównanie położenia biegunów

★ Butterworth

W prawo rośnie częstotliwość, w dół rośnie rząd filtra.

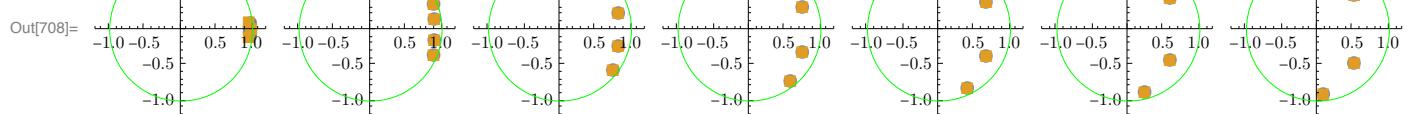
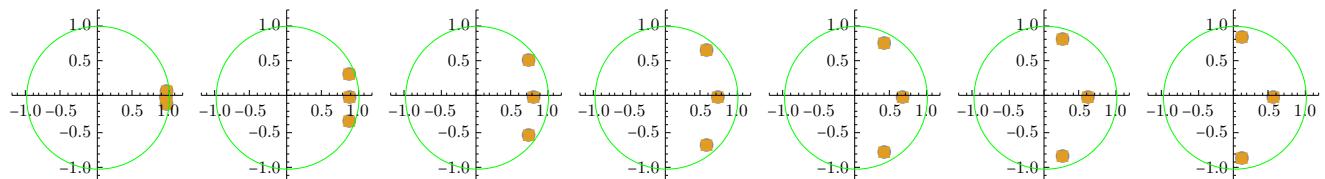
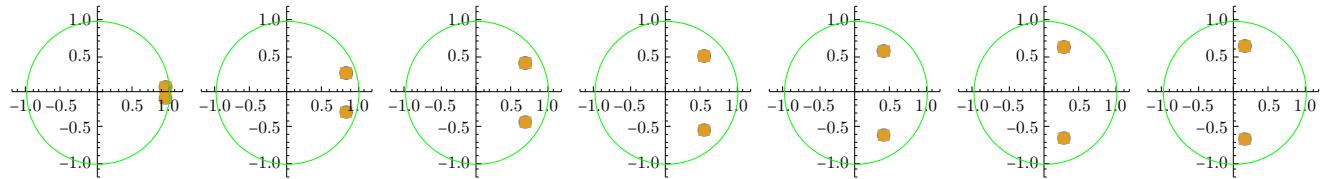
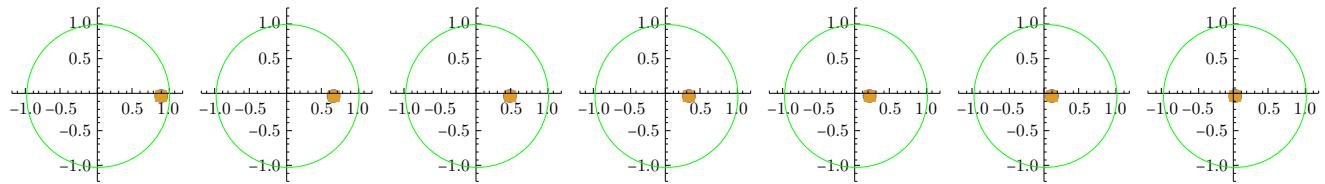
```
In[707]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGbmodels, DGbmodelsDc2}, 2]
```

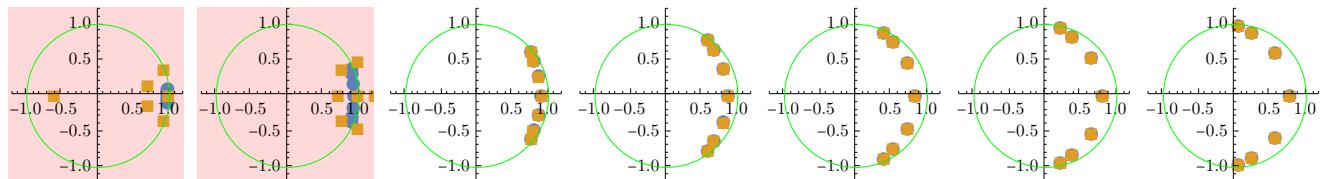
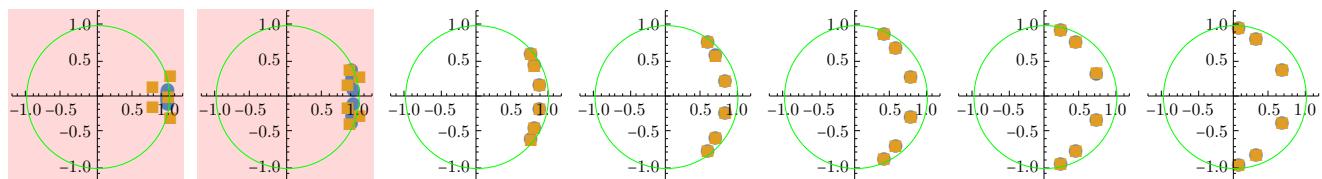
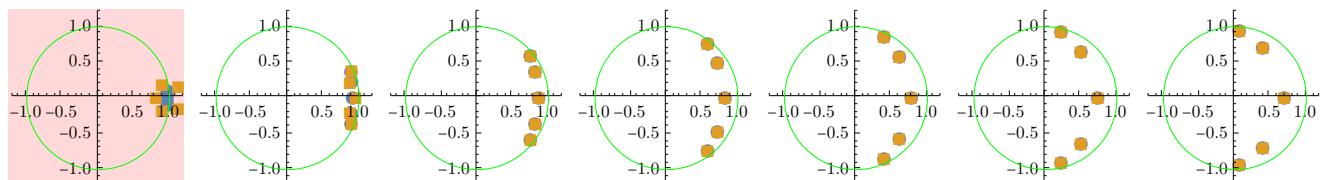




★ Chebyshev 1

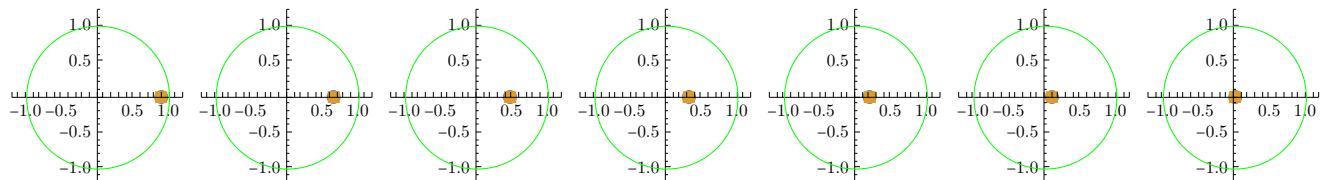
```
In[708]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc1models, DGc1modelsDc2}, 2]
```

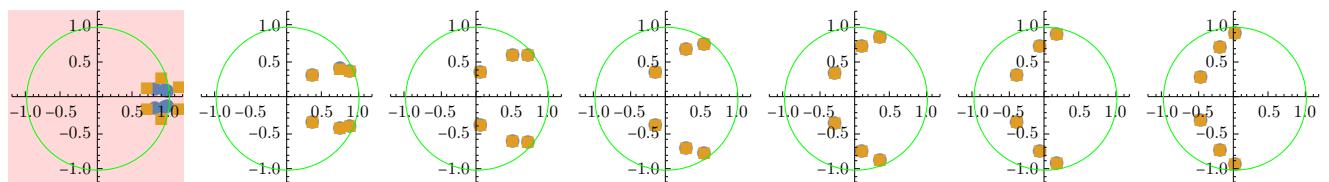
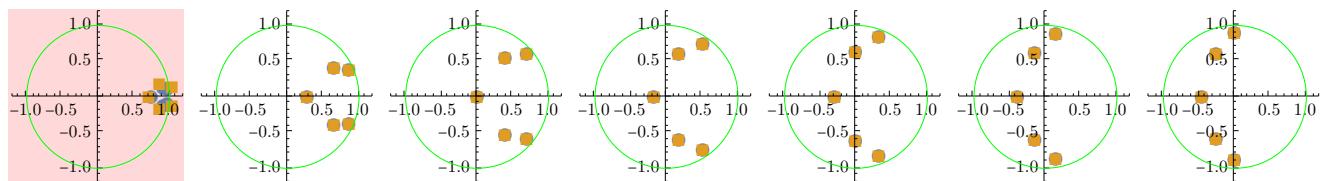
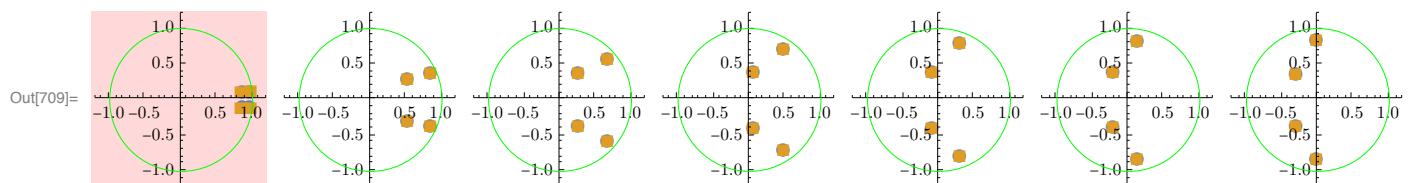
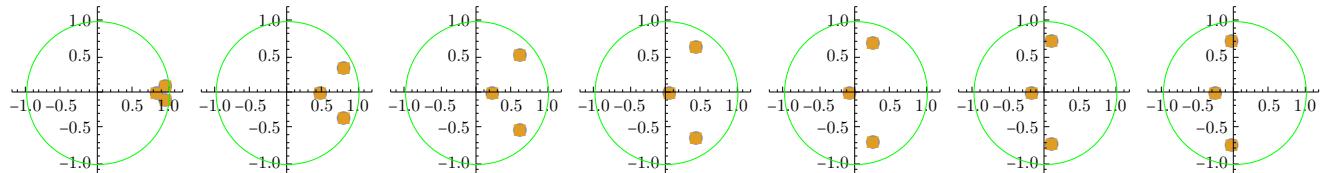
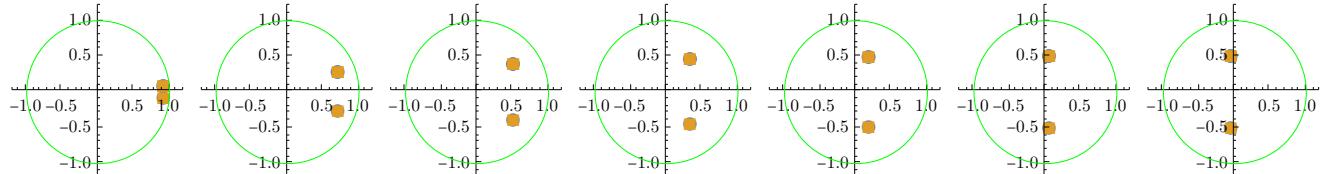


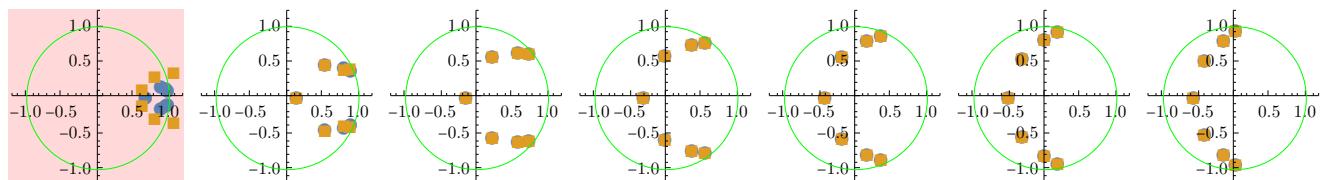


★ Chebyshev 2

```
In[709]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGc2models, DGc2modelsDc2}, 2]
```

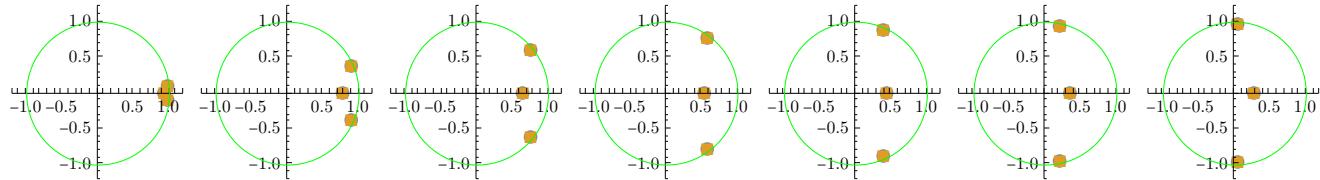
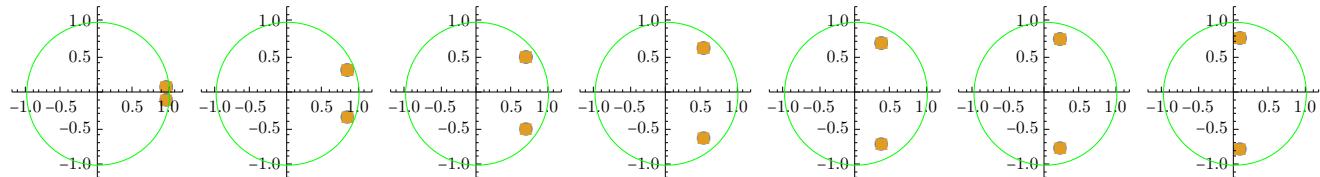
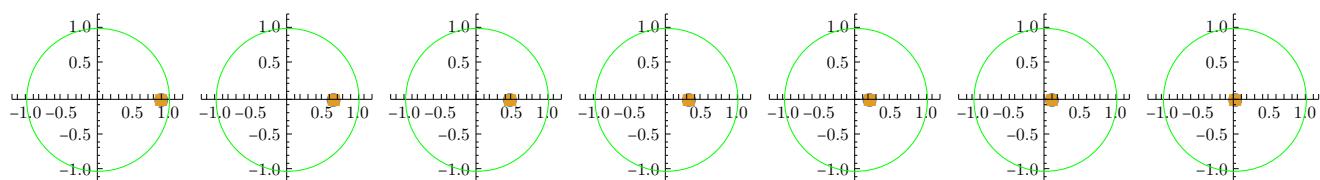


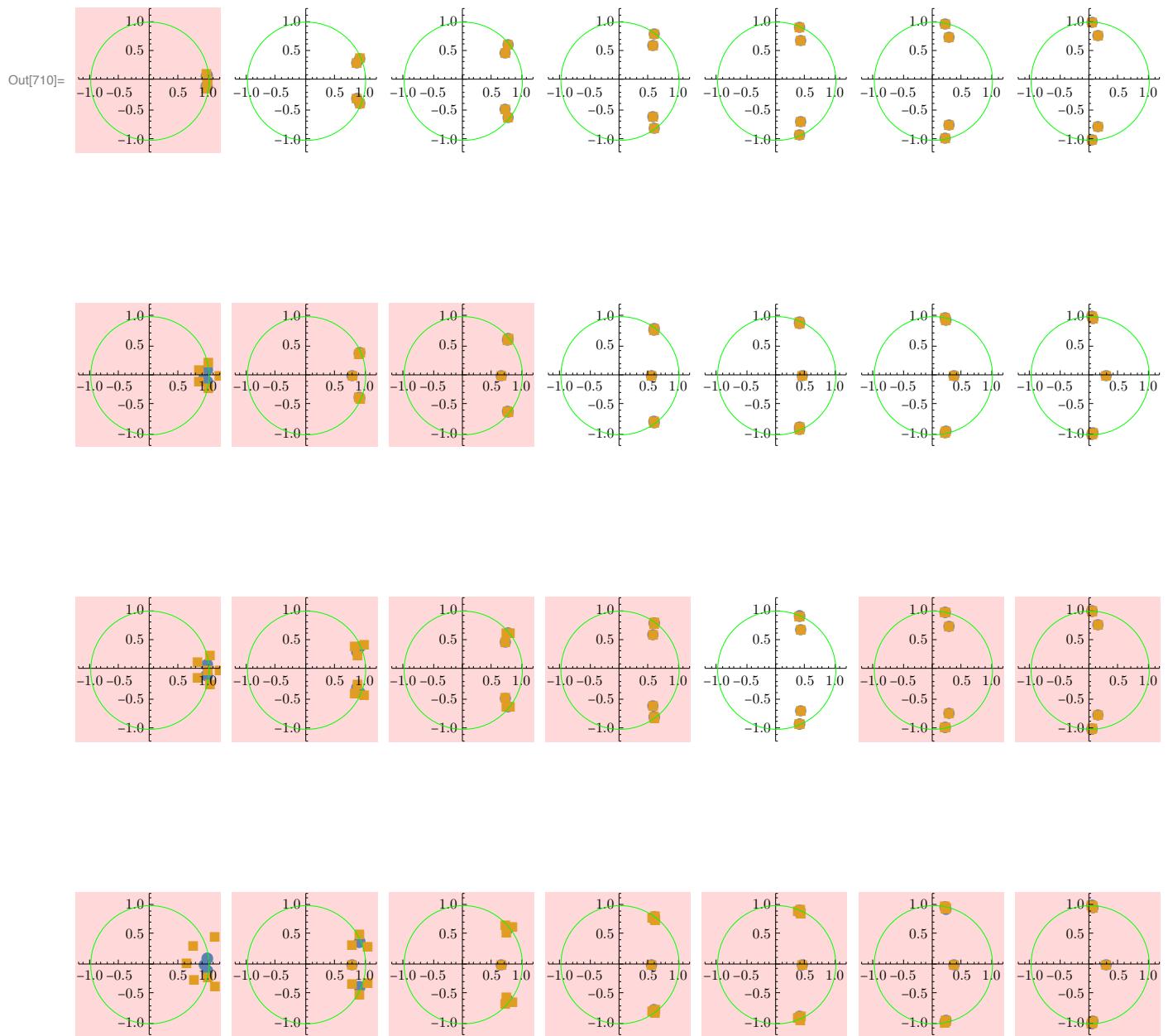




★ Eliptyczne

```
In[710]:= Grid@MapThread[PlotPoles[TransferFunctionPoles[#1][[1, 1]], ExtractPoles[#2]] &,
{DGmodels, DGmodelsDc2}, 2]
```





26

Skompresowana wersja

Zdaje sobie sprawę z tego jak fatalnie się to czyta, zwłaszcza bez spisu treści. Dlatego postanowiłem skompresować najważniejsze wyniki.

Poniżej znajduje się wizualizacja stabilności filtrów, zasada jest ta sama - w dół rośnie rząd, w prawo rośnie częstotliwość graniczna.

A teraz mała incepcja - promień pierścienia jest związany z liczbą bitów. Środkowy okrąg jest 5 bitowy, każdy kolejny 3 bity więcej, okrągów jest 9, zatem ostatni okrąg reprezentuje $5 + 9 * 3 = 5 + 27 = 32$ bity. Kolor niebieski oznacza filtr stabilny, kolor pomarańczowy - niestabilny. Za Zaczynamy!

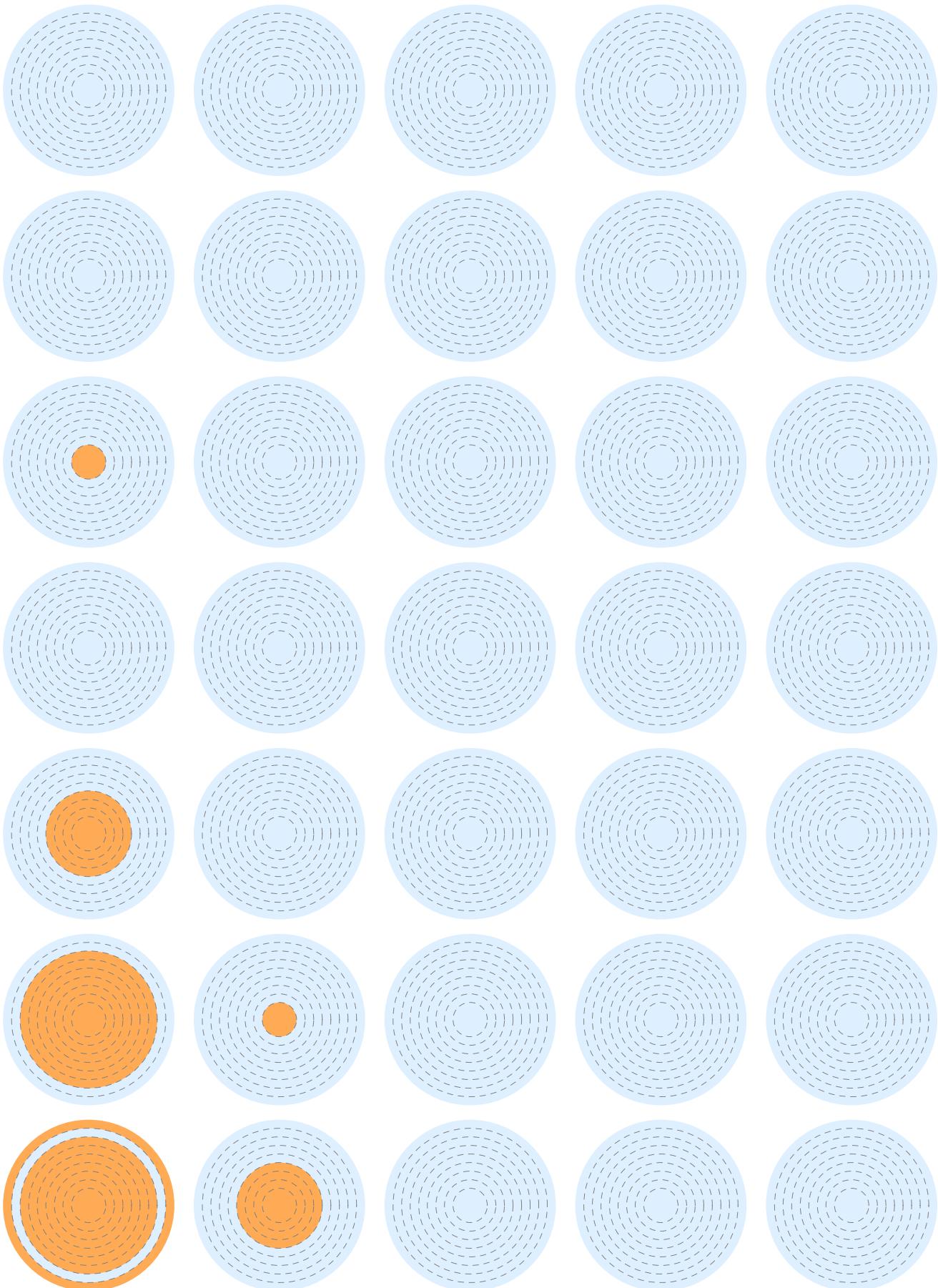
```
In[797]:= SP[p_] := If[Count[p, _? (Abs[#] ≥ 1 &), {1}] > 0, Lighter[Orange], LightBlue]

In[798]:= StableOrNot[p1_, p2_, p3_, p4_, p5_, p6_, p7_, p8_, p9_] := Graphics[{SP[p1], Disk[{0, 0}], EdgeForm[Directive[Dashed, Gray]], SP[p2], Disk[{0, 0}, {0.9, 0.9}], EdgeForm[Directive[Dashed, Gray]], SP[p3], Disk[{0, 0}, {0.8, 0.8}], EdgeForm[Directive[Dashed, Gray]], SP[p4], Disk[{0, 0}, {0.7, 0.7}], EdgeForm[Directive[Dashed, Gray]], SP[p5], Disk[{0, 0}, {0.6, 0.6}], EdgeForm[Directive[Dashed, Gray]], SP[p6], Disk[{0, 0}, {0.5, 0.5}], EdgeForm[Directive[Dashed, Gray]], SP[p6], Disk[{0, 0}, {0.4, 0.4}], EdgeForm[Directive[Dashed, Gray]], SP[p6], Disk[{0, 0}, {0.3, 0.3}], EdgeForm[Directive[Dashed, Gray]], SP[p7], Disk[{0, 0}, {0.2, 0.2}]}]
```

26.1 Butterworth

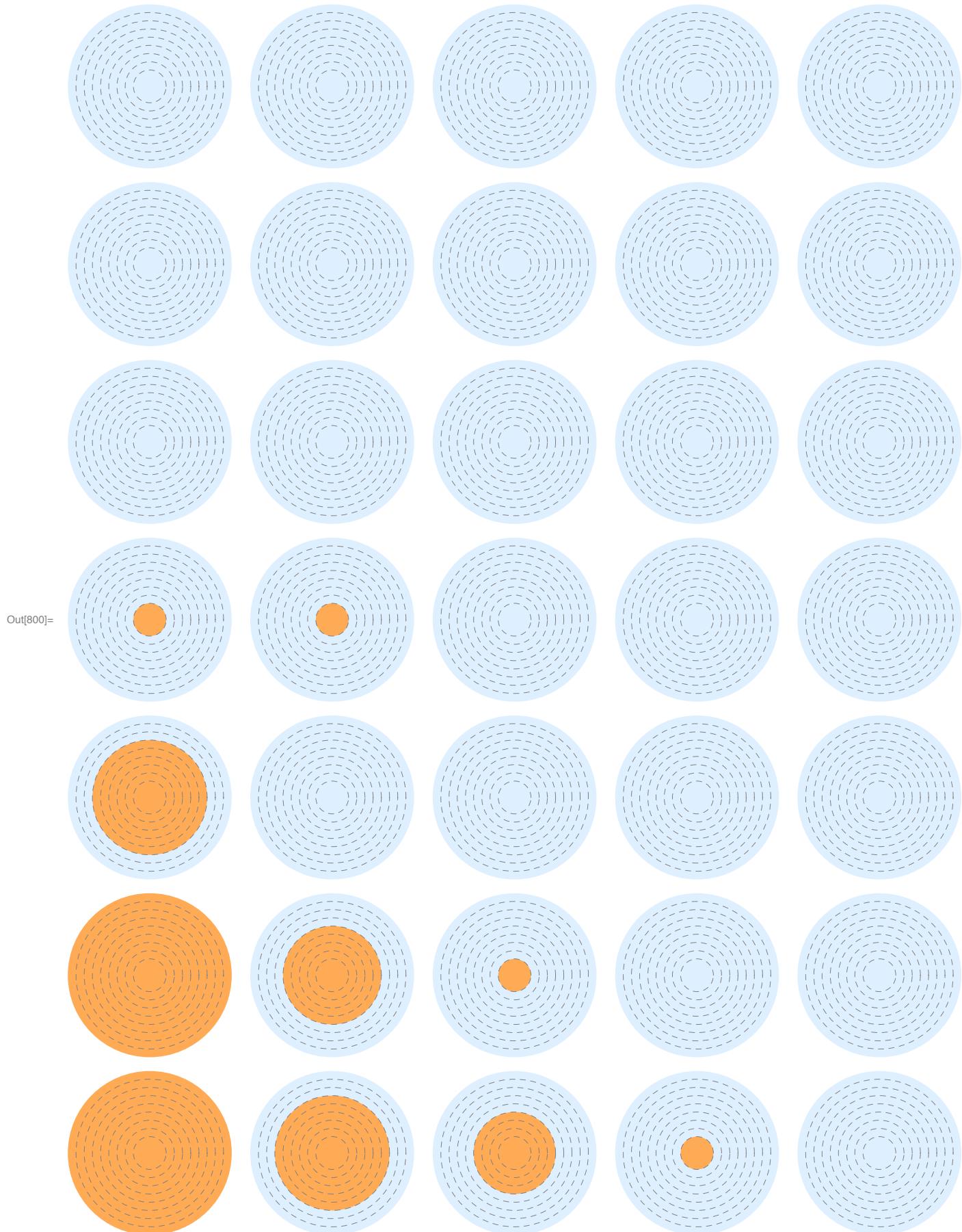
```
In[780]:= Allb = Table[Map[ExtractPoles[DiscretizeModelCoeffs2[#, bits]] &, DGbmodels, {2}], {bits, 5, 32, 3}];

In[799]:= Grid[MapThread[StableOrNot[##9, ##8, ##7, ##6, ##5, ##4, ##3, ##2, ##1] &, Allb, 2], ItemSize → 10]
```



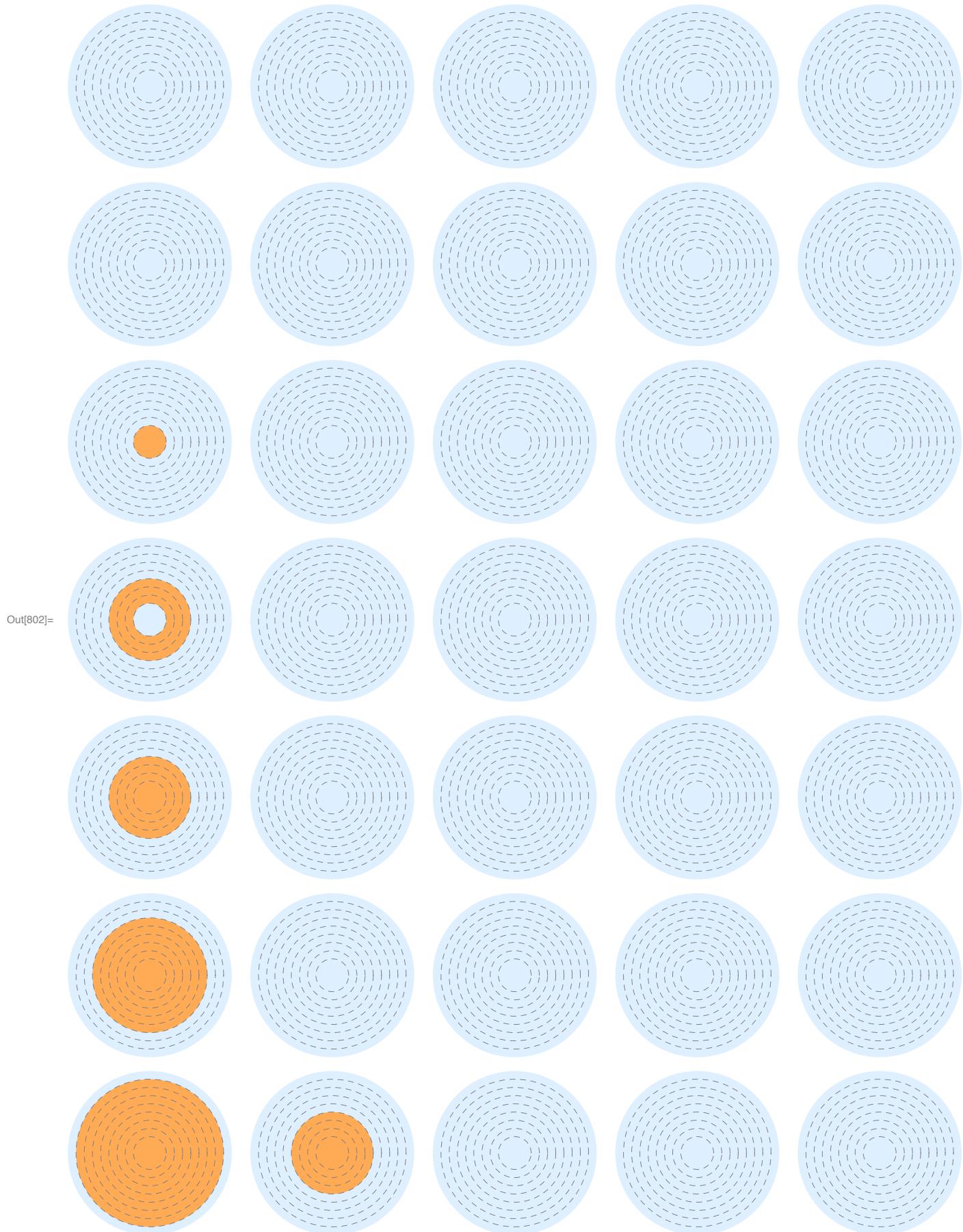
26.2 Chebyshev 1

```
In[783]:= Allc1 = Table[Map[ExtractPoles[DiscretizeModelCoeffs2[#, bits]] &, DGc1models, {2}], {bits, 5, 32, 3}];  
In[800]:= Grid[MapThread[StableOrNot[#9, #8, #7, #6, #5, #4, #3, #2, #1] &, Allc1, 2], ItemSize -> 10]
```



26.3 Chebyshev 2

```
In[801]:= Allc2 = Table[Map[ExtractPoles[DiscretizeModelCoeffs2[#, bits]] &, DGc2models, {2}], {bits, 5, 32, 3}];  
Grid[MapThread[StableOrNot[#, #8, #7, #6, #5, #4, #3, #2, #1] &, Allc2, 2], ItemSize -> 10]
```



26.4 Eliptyczny

```
In[803]:= Alle = Table[Map[ExtractPoles[DiscretizeModelCoeffs2[#, bits]] &, DGmodels, {2}], {bits, 5, 32, 3}];  
Grid[MapThread[StableOrNot[#, #8, #7, #6, #5, #4, #3, #2, #1] &, Alle, 2], ItemSize -> 10]
```

