

# Onderzoek naar knelpunten op de ECT-terminal in het kader van het SMAGIC-project

## Rapportage Fase 1

A.L. Schoute  
Faculteit Informatica  
Universiteit Twente

### 1. Inleiding

Dit rapport bevat het resultaat van een studie gericht op een specifiek verkeersknelpunt bij de DeltaSea-land ECT-terminal. Op dit knelpunt is een kritieke sectie-analyse uitgevoerd met behulp van een analyseprogramma RoadPlan. Dit programma biedt de mogelijkheid om (1) botsingspatronen te berekenen en middels configuratiediagrammen in beeld te brengen en (2) deadlock-vrije uitsluitingzones met bijbehorende claim- en releasepunten te genereren.

Het rapport bevat een beschrijving van de toegepaste methoden en geeft tevens aanbevelingen voor nadere oplossingsstrategieën.

### 2. Botsingsvermijding door wederzijdse uitsluiting.

Botsingen tussen voertuigen zijn te omschrijven als paarsgewijs conflicterende posities van voertuigen, waarbij de ruimten die twee voertuigen innemen overlappen of althans elkaar raken. Dergelijke *botsingconfiguraties* moeten worden afgeschermd door de voortgang van voertuigen op bepaalde momenten te beperken.

Gangbaar is het toepassen van een *blokkering* of *uitsluiting*, waarbij de toegang tot bepaalde *zones* op elk moment slechts aan één voertuig is voorbehouden. De gelijktijdige toegang van meerdere voertuigen tot een zone wordt uitgesloten. De uitsluiting kan worden geëffectueerd door een *claim-release-mechanisme*. Bij een claimpunt is voortgang van een voertuig alleen mogelijk indien de bijbehorende zone “vrij” is. Bij passeren van de claimpunt wordt de zone als “bezet” aangemerkt. Bij een releasepunt wordt de zone weer “vrij”-gegeven. De uitsluitingsvariabele met toestanden “vrij” resp. “bezet” staat in de wereld van de computerproces-synchronisatie bekend als *binair semafoor*<sup>1</sup>. Bij computergestuurde voertuiggeleiding is het aantrekkelijk de term *semafoor* te hanteren vanwege de oorspronkelijke fysieke, plaatsgebonden betekenis. De abstracte betekenis (niet-fysiek, elektronisch synchronisatiemiddel) behoudt immers bij voertuiggeleiding zijn geometrische context.

---

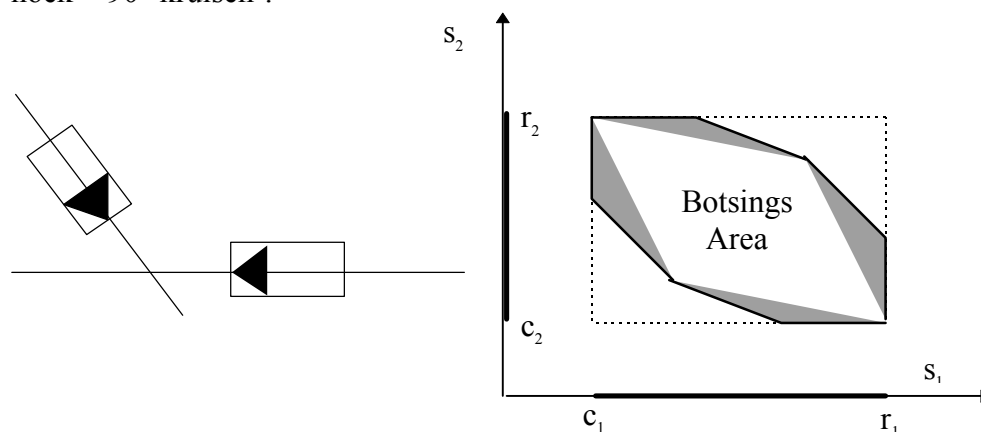
<sup>1</sup>Het er aan gekoppelde zone-begrip is hierbij ontdaan van enige geometrische betekenis en geabstraheerd tot een *kritieke programmasectie*, waarin - meestal vanwege toegang tot een *shared resource* - slechts één proces tegelijk mag executeren.

Generalisatie van het begrip semafoor, alhoewel in dit rapport niet aan de orde, maakt het mogelijk naast uitsluiting ook algemenere vormen van voortgangsbewaking te realiseren<sup>2</sup>.

Veelal wordt uitsluiting gebaseerd op basis van zones die gedefinieerd zijn als oppervlakte-gebieden (bijv. het kruisingsgebied van twee wegen). Claim- en releasepunten liggen op de grens van het gebied.

Zones moeten uiteraard zo gekozen worden dat alle botsingsconfiguraties worden uitgesloten. Een nadeel van “oppervlakte-zones” kan zijn dat behalve botsingsconfiguraties ook botsingsvrije configuraties in grote mate worden uitgesloten. (In het gewone autoverkeer zal een kruispunt meestal vele auto’s kunnen bevatten, d.w.z. gebruik van het kruispunt door hoogstens één voertuig tegelijkertijd is i.h.a. niet acceptabel.) Als voertuigen bovendien niet vrij in de ruimte navigeren maar geleid worden langs vaste banen, zijn de mogelijke botsingsconfiguraties nauwkeuriger af te bakenen. In dit verband is het goed te bedenken dat botsingen niet zozeer bepaald worden door een gebied (elk oppervlak leent zich voor botsingen), als wel door de gezamenlijke positie van elk tweetal voertuigen.

Bij een tweetal geleide voertuigen die zich elk langs een gegeven baan voortbewegen, kunnen we in een zogenaamd *configuratiediagram* botsingsconfiguraties weergeven. In dit diagram is de gezamenlijke toestand te zien afhankelijk van de *baanparameters*  $s_i$  van elk der voertuigen ( $i = 1, 2$ ). De baanparameter representeert de afgelegde weg langs de baan. In onderstaande figuur 1 zijn de botsingsconfiguraties aangegeven van twee rechthoekige voertuigen die rijden over twee rechte banen die elkaar onder een hoek  $> 90^\circ$  kruisen<sup>3</sup>.



**Figuur 1.** Kruisende voertuigen en het bijbehorende configuratiediagram met botsingsarea.

De botsingsarea in het configuratiediagram induceert *kritieke baansecties*  $[c_1, r_1]$  en  $[c_2, r_2]$  op de respectievelijke voertuigbanen. Dit zijn posities van voertuigen die in

<sup>2</sup>Voertuiggeleiding als systeem van automaten (concept, maart 1997), J.J.M. Evers, J.A. Ottjes, M.B. Duinkereken, TRAIL Onderzoeksschool, TUD.

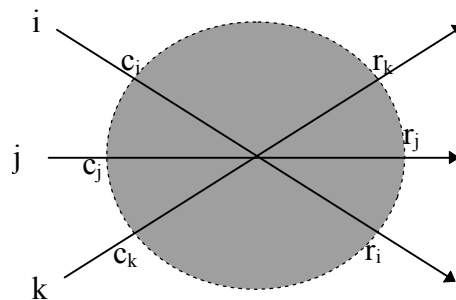
<sup>3</sup> Deadlock-free traffic control with geometrical sections, A.L. Schoute en P.J. Bouwens, Proceedings CSN-94.

combinatie met elkaar tot botsing *kunnen* leiden. Merk op dat de kritieke baansecties de projecties vormen van de botsingsarea op elk der banen.

Als we wederzijdse uitsluiting toepassen op de kritieke baansecties dan worden alle botsingsconfiguraties uitgesloten. De “uitsluitings-zone” wordt gevormd door de combinatie van baansecties  $[c_1, r_1]$  en  $[c_2, r_2]$ , d.w.z. door het *cartesisch product*  $[c_1, r_1] \times [c_2, r_2]$ . We zullen zo’n combinatie van baansecties aanduiden als *cartesische zone*.

Uitsluiting van alle voertuigconfiguraties van een cartesische zone is te realiseren door één binaire semafoor met claims op de posities  $c_1$  en  $c_2$ , en releases op de posities  $r_1$  en  $r_2$ . Een cartesische zone  $[c_1, r_1] \times [c_2, r_2]$  is niet altijd de minimale afdekking van het botsingsarea, zoals te zien is in figuur 1. Ook botsingsvrije configuraties worden, eventueel onnodig, uitgesloten. Sommige botsingsvrije configuraties geven evenwel aanleiding tot *deadlock* bij vooruit rijden, bijv. in het voorbeeld de configuratie  $(s_1, s_2) = (c_1 + \varepsilon_1, c_2 + \varepsilon_2)$ . De rechthoekige afscherming door een cartesische zone voorkomt dan ook in dit geval niet alleen botsing maar tevens mutuele *deadlock*.

Een cartesische zone is, alhoewel mogelijk niet minimaal, in het algemeen nauwkeuriger dan uitsluiting op basis van een oppervlakte zone. We zullen dit aan de hand van een voorbeeld toelichten. Beschouw drie kruisende wegen zoals in figuur 2 weergegeven.

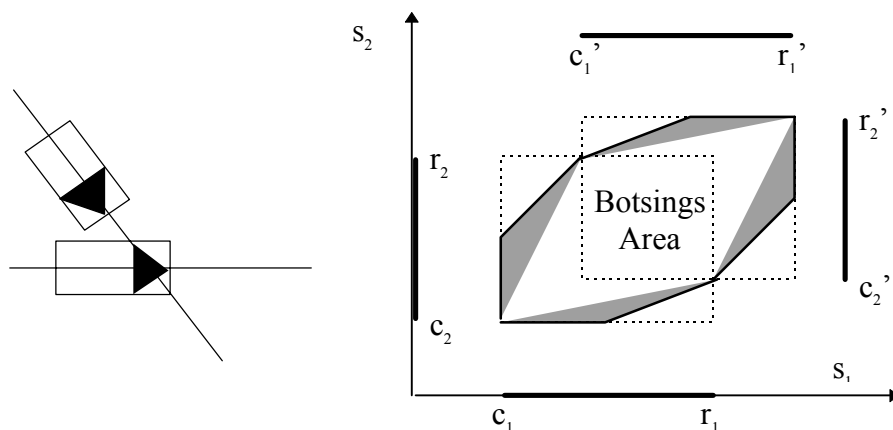


**Figuur 2.** Drie kruisende wegen i, j, k met één uitsluitingsgebied.

Bij bepaling van de oppervlakte-zone wordt in dit geval de straal van het gebied bepaald door de scherpste kruising, d.w.z. de “ontmoetingen” van voertuigen op de banen (i, j) en op de banen (j, k). Uitsluiting tussen twee voertuigen op de banen (i, k) wordt nu onnodig vroeg afgedwongen en tevens wordt de release onnodig laat afgegeven.

Afscherming van het kruispunt door drie cartesische zones voor de baancombinaties (i, j), (j, k) en (i, k), biedt een selectievere uitsluiting. Uiteraard wordt in het laatste geval uitsluiting geregeld door 3 semaforen i.p.v. 1 semafoor bij één oppervlaktezone. Bij gebruik van één oppervlaktezone zijn als het ware de cartesische zones samengevoegd tot een uitsluitingsgebied. Het samenvoegen van cartesische zones vergroot de ruimte van voertuigconfiguraties die niet worden toegelaten. Hierbij zullen i.h.a. botsingsvrije toestanden in de configuratieruimte onnodig worden uitgesloten. Omgekeerd kan het zinvol zijn cartesische zones op te splitsen in meerdere deelzones teneinde de uitsluiting scherper te maken en de kans op blokkering te verkleinen (zie figuur 3). Het beter benaderen van het botsingsarea door kleinere cartesische zones mag evenwel niet tot *deadlock* leiden. We komen hier nog

op terug in paragraaf 4. Als criterium bij het al of niet toepassen van selectieve uitsluiting door meerdere zones kan als maat worden gehanteerd de omvang van het areaal van botsingsvrije toestanden in de configuratieruimte dat bij selectieve uitsluiting wordt toegelaten resp. dat bij niet-selectieve uitsluiting onnodig wordt uitgesloten.



**Figuur 3.** Selectieve afscherming van de botsingsarea bij “scherp” ( $< 90^\circ$ ) kruisende voertuigen door twee cartesische zones.

### 3. De meerbaansbocht bij ECT

De situatie waarvoor specifiek een analyse en oplossingsstrategie wordt gevraagd betreft een 4x4-baansbocht weergegeven in figuur 4. Op de ECT-terminal vormt deze meerbaansbocht een knelpunt bij het oprijden van de voertuigen van de stacking-area naar de kadebanen.

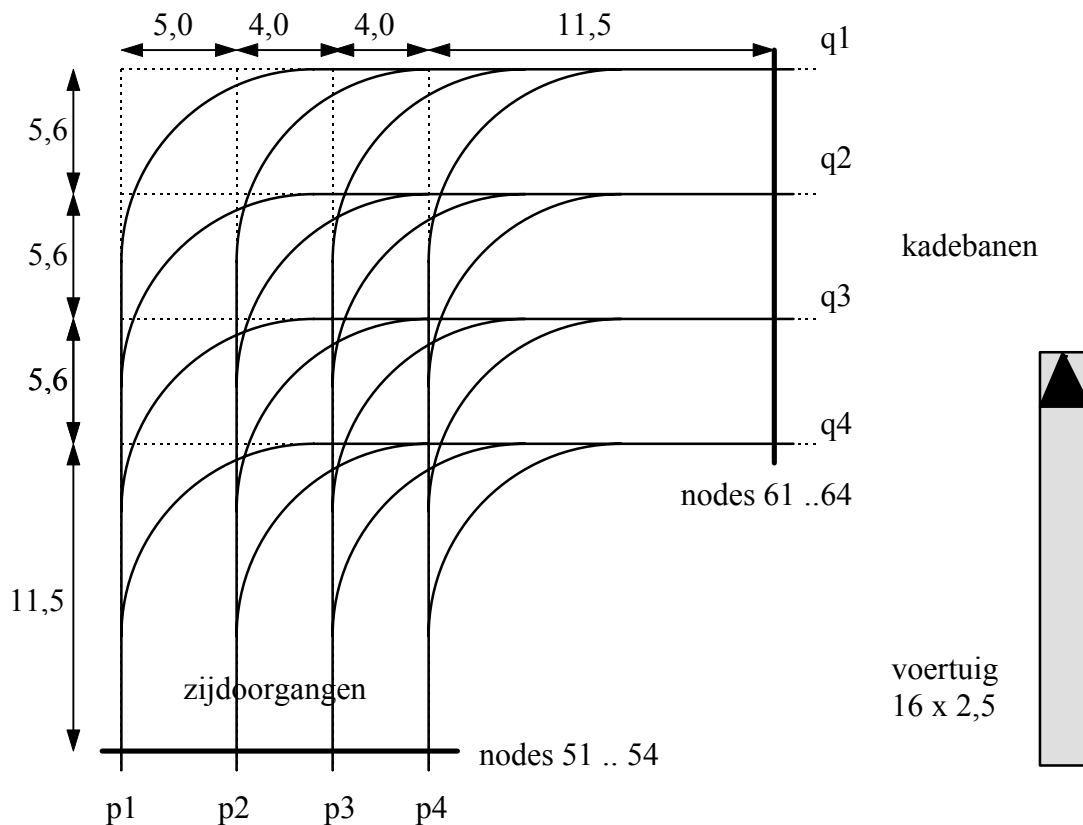
De 4x4-baansbocht is te beschouwen als een wegennet bestaande uit 16 banen, namelijk de verzameling van banen  $B_{ij}$ , waarbij baan  $B_{ij}$  loopt vanaf beginpunt  $p_i$  ( $i = 1, 2, 3, 4$ ) naar eindpunt  $q_j$  ( $j = 1, 2, 3, 4$ ).

De banen zoals deze door de voertuigen worden gereden bestaan (bij benadering) uit een recht beginstuk, daarna een kwart cirkelboog als middenstuk en tenslotte weer een recht eindstuk.

De beschreven 4x4-baansbocht wordt momenteel als één uitsluitingsgebied behandeld. Dit is een zeer groffe vorm van botsingsvermijding die de doorloop van voertuigen beperkt.

De problematiek die samenhangt met het vinden van een meer subtiële oplossing heeft te maken met:

1. de extreme afmetingen van de voertuigen (16 meter lang en 2,5 meter breed), waardoor voertuigen erg ver uitzwaaien en in onderlinge deadlock kunnen geraken.
2. de wenselijkheid om voertuigen niet in de bocht tot stilstand te laten komen teneinde botsingen te voorkomen.



**Figuur 4.** Layout van de 4x4-baansbocht op de ECT-terminal

Bij de analyse van de meerbaansbocht zullen we bekijken hoe twee voertuigen op banen  $B_{ij}$  resp.  $B_{kl}$  elkaar “dwarszitten”. Hieronder valt in principe ook de situatie waarbij twee voertuigen op dezelfde baan rijden ( $ij = kl$ ).

Voor de onderhavige studie zullen we evenwel veronderstellen dat opvolgende voertuigen vanaf hetzelfde beginpunt  $p_i$  elkaar uitsluiten, d.w.z. pas mogen oprijden als de voorganger zijn eindpunt heeft bereikt. Evenzo zullen we veronderstellen dat voertuigen met hetzelfde eindpunt  $q_j$  als bestemming elkaar uitsluiten. Deze veronderstellingen zijn niet essentieel, maar meer van praktische aard: op grond van het gemeenschappelijke begin- of eindstuk is op voorhand in te zien dat het gelijktijdig doorlopen van deze banen gelimiteerd is. We beperken zo het aantal combinaties van banen  $B_{ij}$  en  $B_{kl}$  tot die gevallen waarbij  $i \neq k$  en  $j \neq l$ . Van de oorspronkelijk 120 combinaties ( $16 \text{ over } 2$ ) blijven er dan 72 over.

Bepaalde combinaties van banen zullen op grond van de geometrie vergelijkbare interactiepatronen tussen de daarop rijdende voertuigen te zien geven, bijv. de combinaties  $(B_{11}, B_{22})$  t.o.v.  $(B_{22}, B_{33})$ . Ook de routes die in de praktijk gekozen worden zullen bepaalde baancombinaties meer of minder interessant maken. Het aantal baancombinaties waarvan een analyse gewenst is, is daarmee verder te beperken.

Gezien de systematiek waarmee de analyse evenwel kan worden uitgevoerd en vanwege de beschikbaarheid van een computerprogramma (*RoadPlan*) daarvoor, is de analyse op zich geen beperkende factor. Bepalend is eerder welke simplificaties in de praktische implementatie gewenst zijn. Deze kunnen van de baancombinatie afhankelijk zijn.

#### 4. Analyse-methode

Het analyseren van baancombinaties heeft tot doel om botsingsconfiguraties te onderkennen en cartesische zones te bepalen die deze configuraties afschermen. Botsingsconfiguraties zijn afhankelijk van de geometrie van (1) de banen, (2) de voertuigen en (3) de wijze waarop de voertuigen over de banen rijden.

Een bepaald punt van het voertuig (uitgaande van een symmetrische vorm meestal het midden) wordt als *referentiepunt* genomen. Er wordt verondersteld dat het referentiepunt de baan blijft volgen. Evenzo wordt een bepaalde orientatie van het voertuig als *referentierichting* genomen en wordt verondersteld dat deze referentierichting de raaklijn aan de baan blijft volgen. De positie van een voertuig kan daarmee worden vastgelegd door een *baanparameter* of *baancoördinaat*, d.w.z. de door het referentiepunt afgelegde weg.

Afwijkingen van de baan, zoals een dwarse verplaatsing of een orientatiemarge zullen verdisconteerd worden in de gekozen vorm (oppervlakteprofiel) van het voertuig.

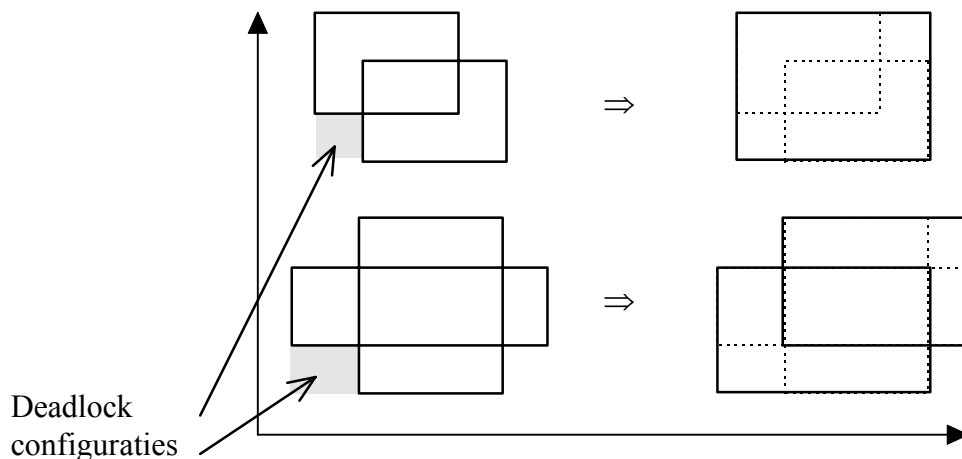
Het theoretische concept van de gevolgde analysemethode is eenvoudig te formuleren:

1. bepaal de botsingsarea in het configuratiediagram van twee voertuigen op de gegeven combinatie van banen;
2. bepaal de cartesische zones die de botsingsarea overdekken met de gewenste mate van nauwkeurigheid.

Een wenselijke randvoorwaarde die extra opgelegd kan worden is:

3. bepaalde cartesische zones zodanig dat mutuele deadlock tussen de voertuigen uitgesloten wordt.

Aan deze laatste randvoorwaarde is eenvoudig te voldoen. Twee cartesische zones  $[c_1, r_1] \times [c_2, r_2]$  en  $[c_1', r_1'] \times [c_2', r_2']$  geven aanleiding tot deadlock als ze op de volgende wijze overlappen:  $c_1 < c_1' < r_1$  en  $c_2' < c_2 < r_2'$ . In figuur 5 is aangegeven hoe mogelijke vormen van deadlock door samenvoeging van cartesische zones dan wel door een andere overdekking opgelost kunnen worden.

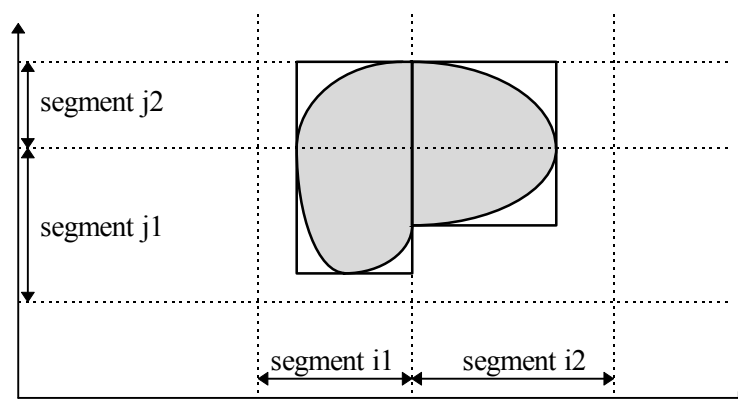


**Figuur 5.** Aanpassing van cartesische zones ter vermijding van mutuele deadlock.

Bij de praktische uitvoering van de botsingsanalyse moeten we uitgaan van eenvoudige veronderstellingen t.a.v. de geometrische vormen. Voertuigvormen die in aanmerkingen komen en tot hanteerbare analyses leiden zijn rechthoeken en cirkels of combinaties daarvan. Banen kunnen worden opgedeeld in *segmenten* die wiskundig eenvoudig te beschrijven zijn, zoals rechte lijnstukken en cirkelbogen. Per baansegment kan een bepaalde voertuigvorm worden aangenomen. Hiermee kunnen afwijkingen worden gecompenseerd.

De methode zoals gehanteerd in het analyseprogramma RoadPlan is als volgt:

1. Voor elke combinatie van twee baansegmenten met bijbehorende voertuigvormen wordt gekeken of er botsing kan optreden en zo ja, wat de minimale en maximale baanparameters zijn van elke samenhangende botsingsarea. De minimale en maximale baanparameters bepalen de kritieke baansecties op de segmenten en daarmee de cartesische zones die de botsingsarea overdekken. Zie figuur 6.
2. Cartesische zones worden in principe alleen samengevoegd voorzover dit mogelijk is zonder de overdekking te vergroten. Deadlockconfiguraties en niet bereikbare configuraties worden bij dit *merging-proces* wel overdekt.



**Figuur 6.** Segmentgewijze analyse van baancombinaties.

De mogelijkheden van het RoadPlan-analyseprogramma zijn momenteel als volgt:

- baansegmenten zijn beperkt tot rechte lijnstukken; aan cirkelbogen wordt gewerkt.
- elke combinatie van baansegmenten draagt bij aan hoogstens één cartesische zone, d.w.z. fijnchaligheid van uitsluiting wordt bereikt door een baan te verdelen in een groot aantal segmenten
- het omgekeerde, d.w.z. groffere uitsluiting, kan bereikt worden door grote segmenten te nemen of door instelling van een *zoneGranularity*-parameter.
- de *zoneGranularity* bepaalt hoever een cartesische zone kan worden verruimd (door releasepunten te verschuiven), zodat cartesische zones - al of niet gedeeltelijk - worden samengevoegd.
- de analyse houdt (nog) geen rekening met het verspringen van de orientatie bij de overgang tussen segmenten, d.w.z. met het draaien van voertuigen op deze overgangen; hiervoor moet worden gecompenseerd door geschikte keuze van de voertuigvorm.
- de analyse levert een complete, geordende lijst van alle claim en releasepunten langs het wegennet met hun posities en de semaforen waar ze op werken.

## 5. Modelleren van de meerbaansbocht met RoadPlan

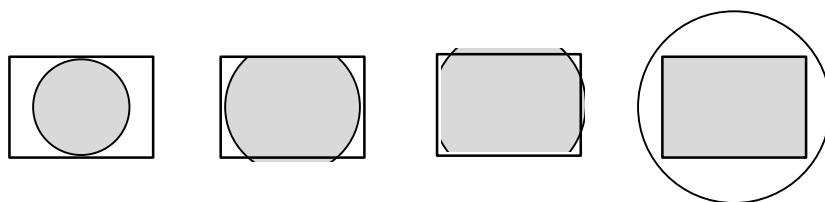
De huidige mogelijkheden van RoadPlan tot automatische analyse zijn beperkt tot rechte lijnsegmenten, zodat de meerbaansbocht zo goed mogelijk moet worden gemodelleerd door rechte lijnstukken. Hierbij hanteren we een *veilige* benadering, d.w.z. we zorgen voor een conservatieve uitsluiting door afwijkingen van de werkelijke baan te compenseren door de voertuigvorm groter te nemen. Ook deviaties die in de praktijk voorkomen of gewenste veiligheidsmarges kunnen op deze wijze meegenomen worden.

Een gemodelleerde baan is een veilige benadering van de werkelijke situatie als de *hull-curve* van het model alle werkelijk voorkomende voertuigposities afdekt. De *hull-curve* is hierbij het gebied “rond” de baancurve dat door het voertuig wordt “doorsneden”.

De layout van het wegennet wordt in RoadPlan ingevoerd middels een “road-map”-file (met extensie *rm*). Banen (of paden in RoadPlan terminologie) bestaan uit lijnsegmenten tussen een lijst van punten ( $x_i, y_i$ ). Ze worden in de file beschreven door de volgende constructie:

```
PATH
  POINT X1, Y1
  POINT X2, Y2
  ...
  POINT Xn, Yn
ENDPATH
```

Na elk punt kan hierbij een (andere) voertuigvorm worden gespecificeerd, dan wel worden aangepast door middel van de parameters DIAMETER, WIDTH en LENGTH. De voertuigvorm op een lijnsegment wordt bepaald door de actuele waarden van deze parameters bij het eindpunt van dat segment. De resulterende voertuigvorm is de doorsnede van een cirkelschijf met de gegeven diameter en een rechthoek met de gegeven lengte en breedte. Hiermee zijn verschillende vormen vast te leggen zoals getoond in figuur 7.



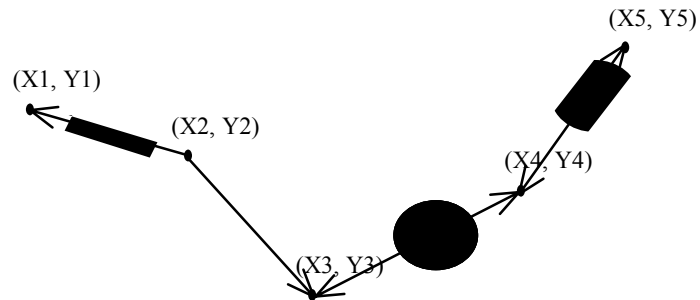
**Figuur 7.** Verschillende voertuigvormen te specificeren door middel van de parameters DIAMETER, WIDTH en LENGTH.

Onderstaand voorbeeld correspondeert met de baan in figuur 8 waarbij de voertuigvormen zijn aangegeven.

```
PATH
  POINT X1, Y1
  DIAMETER = 15.21 LENGTH = 15 WIDTH = 2.5
  POINT X2, Y2
```



POINT X3, Y3  
 DIAMETER = 15 WIDTH = 15  
 POINT X4, Y4  
 LENGTH = 15.21 WIDTH = 5  
 POINT X5, Y5  
 ENDPATH



**Figuur 8.** Voorbeeld van een gemodelleerde baan met voertuigvorm verandering.

Oriëntatieafwijkingen op de baan ter grootte van  $\pm \Delta\alpha$  zijn voor een rechthoekig voertuig met lengte L en breedte W te compenseren door de vormparameters als volgt te kiezen:

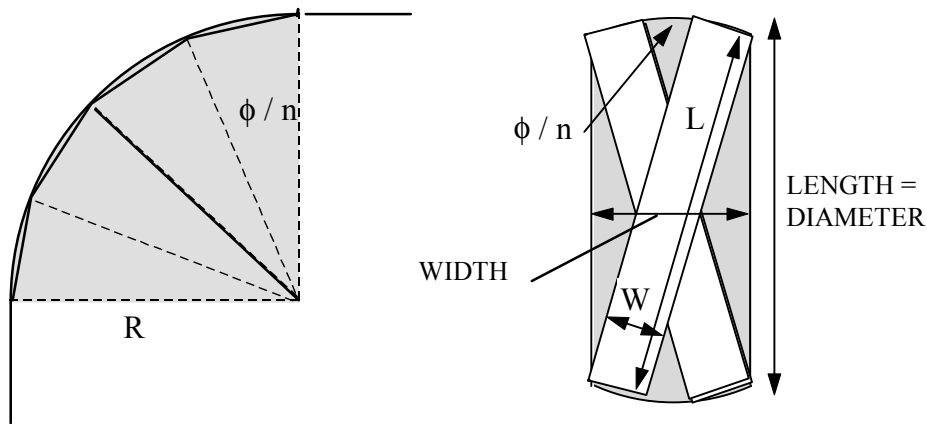
$$\text{WIDTH} = W \cdot \cos\Delta\alpha + L \cdot \sin\Delta\alpha, \text{ LENGTH} = \text{DIAMETER} = \sqrt{(W^2 + L^2)}$$

Uitwijkingen  $\pm \Delta u$  dwars op de baan kunnen worden meegenomen door de voertuigbreedte te vergroten:  $\text{WIDTH} = W + 2\Delta u$

De benadering van een cirkelboog met kromtestraal R en openingshoek  $\phi$  door n segmenten (zie figuur 9) vereist de volgende aanpassing van een rechthoekig voertuig:

$$\begin{aligned} \text{WIDTH} &= W \cdot \cos\Delta\alpha + L \cdot \sin\Delta\alpha + 2R(1 - \cos\Delta\alpha) \text{ met } \Delta\alpha = \Delta\phi / 2n \\ \text{LENGTH} &= \text{DIAMETER} = \sqrt{(W^2 + L^2)} \end{aligned}$$

De oriëntatieafwijking op een benaderend bochtsegment loopt van  $-\Delta\phi / 2n$  naar  $+\Delta\phi / 2n$ , zodat we een oriëntatiemarge veronderstellen van  $\pm \Delta\alpha$  met  $\Delta\alpha = \Delta\phi / 2n$ . Ook de uitwijking  $\Delta u = R(1 - \cos\Delta\alpha)$  dwars op de baan wordt bepaald door deze ‘‘halve openingshoek’’  $\Delta\alpha$ .



**Figuur 9.** Benadering van een cirkelboog met draaicirkel  $R$  en openingshoek  $\phi$  door  $n$  lijnsegmenten en bijbehorende aanpassing van een rechthoekig voertuig.

Voor de 4x4-baansbocht op de ECT-terminal is de draaicirkel ongeveer  $R = 11.5$  meter. De noodzakelijke aanpassing van de breedte bij een benadering door  $n$  segmenten is aangegeven in onderstaande tabel.

$n$	$\Delta\alpha$	WIDTH
1	$\pi / 4$	19.82
2	$\pi / 8$	10.18
4	$\pi / 16$	6.02
8	$\pi / 32$	4.17
16	$\pi / 64$	3.31

In het RoadPlan-programma is een toevoeging gemaakt om automatisch een cirkelboog te benaderen door lijnsegmenten, waarbij tevens de voertuigvorm wordt aangepast. Bijv. een rechterbocht tussen twee punten  $(X1, Y1)$  en  $(X2, Y2)$  met openingshoek  $45^\circ$  en benaderd door 8 lijnsegmenten wordt als volgt gegenereerd:

```
POINT      X1, Y1
RTURN      45
NPOINT 8   X2, Y2
```

De listing van de road-map-file *ectmodel.rm* waarmee de analyse van de meerbaansbocht is uitgevoerd, is te vinden in Appendix A. Teneinde de aangepaste voertuigbreedte voldoende klein te houden worden bochten benaderd door 16 lijnsegmenten.

## 6. Resultaten gegenereerd door RoadPlan

Botsingsconfiguraties en de generatie van cartesische zones kunnen met RoadPlan interactief worden beoordeeld. Daarnaast levert RoadPlan een zogenaamde *revised-road-map-file* (met extensie *.rrm*), waarin als claim- en releasepunten worden opgeleverd.

Interactieve inspectie laat voor elke baancombinatie het volgende zien:

- Het configuratiediagram met daarin de botsingsarea als deze bestaat. De nauwkeurigheid waarmee de botsingsarea berekend wordt, is instelbaar met de parameter *collisionGranularity*.
- De overdekking van de botsingsarea door cartesische zones. Er is een parameterinstelling “single-zone” ingevoerd waarmee de botsingsarea door één cartesische zone wordt overdekt. Indien deze instelling is uitgeschakeld bepaalt RoadPlan automatisch een bepaalde zone-overdekking. De grootte van de zones is hierbij afhankelijk van de segmentatie van de banen en van de instelling van de “merging”-parameter: *zoneGranularity*.
- Segmentgrenzen zijn in de botsingsarea weergegeven als witte stippellijnen.

Uit het configuratiediagram van een baancombinatie zijn een aantal zaken direct af te leiden.

- de aan- of afwezigheid van botsingstoestanden
- botsingsvrijheid op de randen van het diagram, d.w.z. de botsingsvrijheid van de voertuigen op hun begin- resp. eindposities.
- de relatieve voortgang die voertuigen langs hun banen zouden moeten maken om botsingsvrij te passeren (d.w.z. hun onderlinge snelheidsprofiel).
- de afdekking door cartesische zones: hoe meer zones, hoe beter de nauwkeurigheid, maar hoe intensiever het claim-release-mechanisme.

Bij de beoordeling van het ECT-model beperken we ons (zoals in paragraaf 3 aangegeven) tot baancombinaties zonder gemeenschappelijke begin- en eindpunten. (Hiervoor is in RoadPlan een parameter ingevoerd: *DiffPathCombinations*, die interactief gezet kan worden.)

We onderscheiden verder drie typen van botsingsconfiguraties die in de gespecificeerde meerbaansbocht voorkomen:

Type1. De banen kruisen elkaar (bijv.  $B_{13}$  en  $B_{42}$ ) en de botsingsarea is een samenhangend gebied.

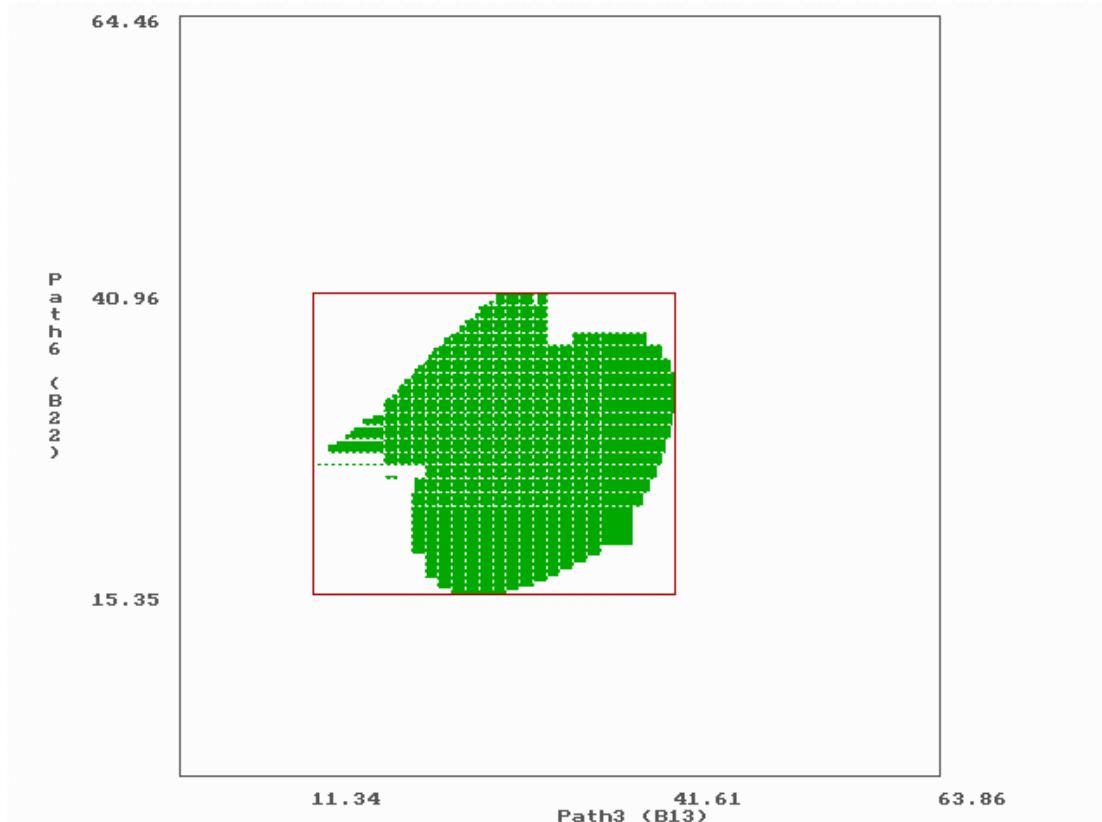
Type2. De banen lopen min of meer parallel (bijv.  $B_{11}$  en  $B_{22}$ ). Dit levert twee gescheiden gebieden waar botsingsconfiguraties kunnen voorkomen. Een voertuig in de binnenbocht hindert namelijk door uitzwaaien mogelijk aan twee kanten voertuigen op de buitenbaan.

Type3. De banen kruisen niet maar lopen “uiteen” (bijv.  $B_{11}$  en  $B_{23}$ ) of komen “bijeem” (bijv.  $B_{11}$  en  $B_{32}$ ). Dit levert één gebied waar botsingsconfiguraties kunnen voorkomen. Het uitzwaaien is slechts aan één kant kritisch.

Overigens is de layout van de meerbaansbocht zo gekozen dat botsingen in sommige gevallen “marginaal” optreden, d.w.z. net niet of net wel. In combinatie met de benadering van de banen door rechte lijnsegmenten, levert dit versnipperde botsingsarea’s op. Het kan zijn dat sommige botsingsarea’s bij nauwkeurigere benadering niet blijken voor te komen. De layout is ook asymmetrisch hetgeen verschillende botsingspatronen oplevert.

We zullen de verschillende botsingsconfiguraties afzonderlijk bespreken.

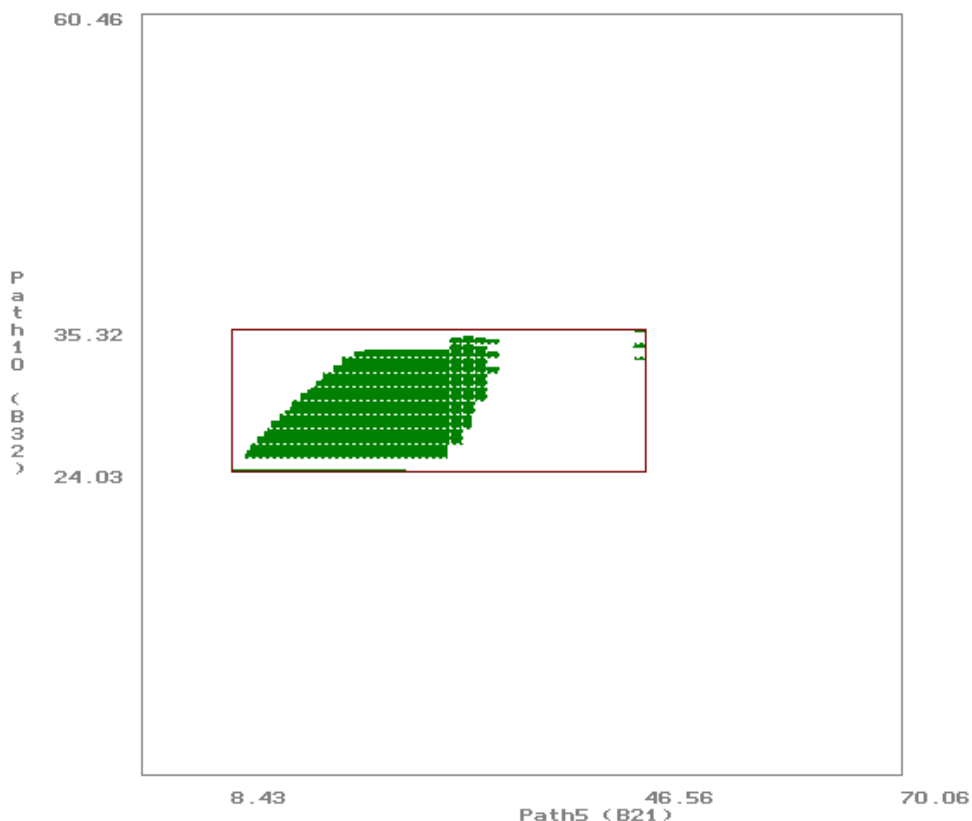
Type1. Er zijn 36 baancombinaties  $B_{ij}$  en  $B_{kl}$  die elkaar kruisen, d.w.z. waarbij  $i < k$  en  $j > l$ . Een voorbeeld configuratiediagram is te zien in figuur 10. In het diagram is de botsingsarea door één enkele cartesische zone afgeschemd. De positie en omvang van de (enkelvoudige) cartesische zones bij de verschillende kruisingen varieert uiteraard. De lengtes van de kritieke baansecties liggen in de orde van 20 à 30 meter. Deze lengte bepaalt de maximale wachttijd die een voertuig vanwege botsingsvermijding kan oplopen.



**Figuur 10.** Conguratiediagram van kruisende banen B13 en B22. De botsingsarea is “donker” aangegeven en ingesloten in één cartesische zone. De lengte van de banen en de baancoördinaten van de claim- en releasepunten staan langs de assen vermeld.

Type2. Alleen direct naast elkaar gelegen “parallele” banen, zoals  $B_{11}$  en  $B_{22}$ , veroorzaken botsingsconfiguraties. In totaal zijn er 9 van dergelijke baancombinaties ( $1 \leq i \leq 3$ ,  $1 \leq j \leq 3$ ,  $k = i + 1$  en  $l = j + 1$ ). Een voorbeeld configuratiediagram is te zien in figuur 11. Daaruit is af te leiden dat botsingen ontstaan door het uitzwaaien van een voertuig op de binnenbocht (middengedeelte van de verticale as). Dit voertuig raakt een voertuig op de buitenbaan dat zich bevindt vóór de bocht op de zijdoorgang (linker gedeelte horizontale as) of ná de bocht op de kadebaan (rechter gedeelte horizontale as). Aangezien de afstand tussen de kadebanen ruimer gekozen is dan tussen de zijdoorgangen, is er slechts een minieme botsingsarea met een voertuig op de kadebaan. Vermoedelijk zullen voertuigen elkaar daar rakelings kunnen passeren en worden botsingsconfiguraties gedetecteerd vanwege de conservatieve benadering van de bocht. Dit is na te gaan door scherpere benaderingen te nemen.

De lengte van de kritieke baansecties bij uitsluiting door een enkelvoudige cartesische zone is in de orde van ruim 35 meter op de buitenbaan en 9 meter op de binnenbaan.

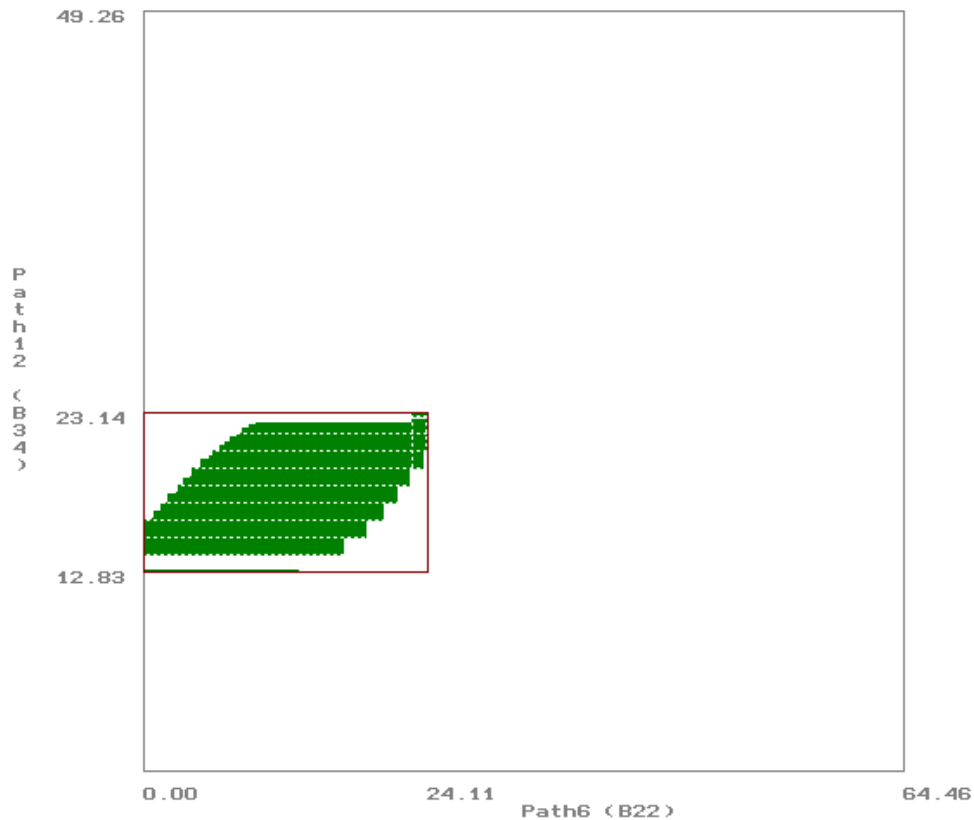


**Figuur 11.** Conguratiediagram van naast-elkaar-gelegen, parallelle banen B21 en B32 (baancombinatie type 2).

Type3. “Toelopende” baancombinaties, zoals  $B_{11}$  en  $B_{32}$ , waarbij voertuigen oprijden vanaf niet-naast-elkaar-gelegen zijdoorgangen naar wel-naast-elkaar-gelegen kadebanen, blijken geen botsingen op te leveren. Dit heeft te maken met de ruimere spatiëring van de kadebanen.

“Uiteenlopende” baancombinaties, zoals  $B_{11}$  en  $B_{23}$ , waarbij voertuigen oprijden vanaf naast-elkaar-gelegen zijdoorgangen, geven wel botsingsconfiguraties als gevolg van de krappe spatiëring van de zijdoorgangen. Deze spatiëring is ook ongelijk zodat de botsingsarea bij ( $B_{11}$ ,  $B_{23}$ ) minder groot is dan bij bijv. ( $B_{22}$ ,  $B_{34}$ ). Het configuratiediagram van de laatste baancombinatie is weergegeven in figuur 12.

De lengte van de kritieke baansecties bij deze baancombinaties is in de orde van 24 meter voor de buitenbaan en 10 meter voor de binnenbaan.



**Figuur 12.** Conguratiediagram van naast-elkaar-beginnende, “uiteenlopende” banen B22 en B34 (baancombinatie type 3)

Uitgaande van een voertuigsnelheid van 3 m/sec. is de typische maximale wachttijd te bepalen die kan optreden als gevolg van uitsluiting bij toepassing van een enkelvoudige cartesische zone. In onderstaande tabel is deze maximale wachttijd voor de verschillende typen baancombinaties bij benadering aangegeven. Vertraging die een voertuig oploopt vanwege stoppen en starten is hierbij niet meegenomen. De wachttijd bestaat alleen uit de tijd die het andere voertuig nodig heeft om de kritieke baansectie op volle snelheid te verlaten.

	Voertuig (beginnend) in buitenbaan	Voertuig (beginnend) in binnenbaan
Kruisende banen (type1)	7 á 10 sec.	7 á 10 sec.
Naast-gelegen, parallelle banen (type2)	3 sec.	7 sec.
Naast-gelegen, uiteenlopende banen (type 3)	3.3 sec.	8 sec.

**Tabel.** Orde van grootte van de maximale wachttijd door enkelvoudige zone-uitsluiting (exclusief stop- en startvertraging).

De gepresenteerde resultaten gaan uit van een enkelvoudige cartesische zone die per baancombinatie de botsingsarea afschermt. Met het RoadPlan-programma is een scherpere afscherming door meerdere zones te berekenen. De resulterende

overdekking door cartesische zones is afhankelijk van de keuze van de baansegmenten en van de parameter *zoneGranularity*. Een nadeel op dit moment is dat de gebruiker deze fijnere indeling niet goed kan sturen. Bovendien is het de vraag of de winst bij fjnshalige afscherming door meerdere zones opweegt tegen de grotere complexiteit. Op dit punt is verder onderzoek nodig. De gebruiker van RoadPlan zou interactief moeten kunnen aangeven hoe cartesische zones fijner opgesplitst moeten worden. Uitgaande van enkelvoudige zones zijn de claim-en releasepunten die het RoadPlan-programma genereert voor het ECT-model te vinden in de file *ectmodel.rrm*, waarvan een gedeelte opgenomen is in appendix B. Het aantal cartesische zones correspondeert met de 54 beschouwde baancombinaties die een botsingsarea opleveren. De nummering van de semaforen, die als parameter bij de claims en releases voorkomen, heeft geen directe relatie met de baancombinatie en hangt samen met de interne verwerkingsvolgorde van RoadPlan.

## **7. Aanbevelingen voor nadere oplossingstrategieën**

Voor het analyseren van knelpunten en het bepalen van oplossingsstrategieën is een goed inzicht in het optreden en de vorm van botsingsarea's tussen alternatieve routes onontbeerlijk. De mogelijkheid om automatisch configuratiediagrammen te berekenen en interactief te beoordelen is dan ook een krachtig hulpmiddel.

Een configuratiediagram met botsingsarea biedt meer dan alleen een methode om de plaatsing van uitsluitingszones en claim- en releasepunten te bepalen.

We zullen dit aan de hand van diverse aspecten en de bijbehorende aanbevelingen duidelijk maken.

### **1. Optimalisatie van de layout**

De efficiëntie van de meerbaansbocht is uiteraard gebaat bij minimalisatie van de botsingsarea's. In het bijzonder zouden niet-kruisende banen geen interferentie tussen voertuigen moeten opleveren. Uit de aard van de botsingsarea's kan worden afgeleid dat de gespecificeerde layout van de meerbaansbocht hier al gedeeltelijk aan voldoet. Er valt veel voor te zeggen om de spatiëring van de layout zodanig aan te passen dat alle niet-kruisende banen zonder onderlinge botsingsconfiguraties doorlopen kunnen worden.

### **2. Bepaling van botsingsmarges**

Aansluitend op het bepalen van onderling botsingsvrije routes, kan bekeken worden welke marges er zijn door voertuigvormen te vergroten en het effect op het optreden van botsingsarea's te analyseren.

### **3. Blokkeringsvrije stopposities**

Als een voertuig stopt binnen een kritieke baansectie van een cartesische zone betekent dit dat een ander voertuig kan worden geblokkeerd op de daarmee verbonden andere kritieke baansectie. Het is in dit verband van belang te weten of de beginpositie dan wel de eindpositie van een baan zich al of niet in kritieke secties bevindt, d.w.z. of

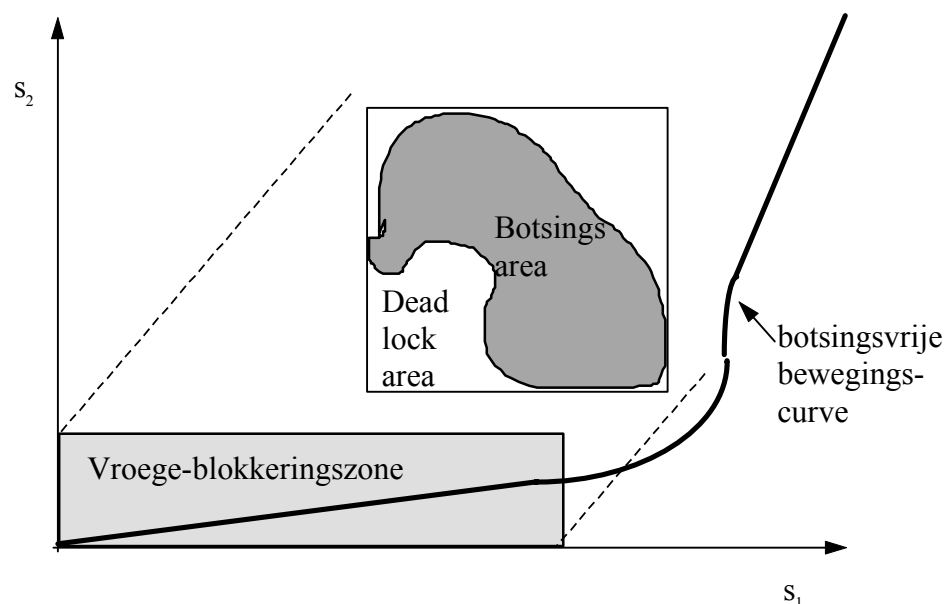
de botsingsarea's zich tot de randen van het configuratiediagram uitstrekken. Wat betreft het ECT-model is af te leiden waar de begin- en eindnodes van de meerbaansbocht gepositioneerd moeten worden zodat ze voor geen enkele andere baan blokkerend werken.

#### 4. Detectie van mutuele deadlock

De vorm van de botsingsarea is bepalend voor het kunnen optreden van deadlock. Deadlock kan ontstaan als de "onderranden" (d.w.z. de botsingsconfiguraties met minimale  $s_1$  resp.  $s_2$ ) monotoon afnemend zijn in een van de baancoördinaten (zie figuur 13). Bij het ECT-model is deadlock - als gevolg van het uitzwaaien van voertuigen in de bocht - eenvoudig visueel te herkennen in de configuratiediagrammen.

#### 5. Aanpassing snelheidsprofiel

Een aantrekkelijk alternatief voor het "stop-and-go"-rijgedrag waarmee momenteel botsingen worden vermeden, is het plannen van een aangepast snelheidsprofiel. Uit het configuratiediagram is zeer direct af te leiden welke relatieve snelheden de voertuigen moeten aannemen om botsingsvrij te passeren. Een relatief snelheidsprofiel bepaalt de simultane bewegingscurve in het diagram (zie figuur 13). Uiteraard moet bij het plannen van snelheden rekening worden gehouden met meerdere voertuigcombinaties. Het is aan te bevelen nader onderzoek uit te voeren ten aanzien van deze alternatieve benadering.



**Figuur 13.** Configuratiediagram ter illustratie met deadlockconfiguraties, een botsingsvrije, relatieve bewegingscurve en vroege-blokkeringszone.



## 6. Non-stop-garantie door “vroege blokkering”.

Het stoppen van voertuigen halverwege de meerbaansbocht is om meerdere redenen onaantrekkelijk. Veelal zal een stoppositie ergens midden op de baan de doorgang op andere banen blokkeren. Verder zal een eenmaal gekozen route niet meer teruggedraaid kunnen worden. Het is daarom wenselijk voertuigen - indien nodig - vroegtijdig te laten wachten op blokkeringsvrije posities waarbij tevens de vrijheid van baankeuze nog open blijft. Na vertrek uit zo'n stoppositie zou een *non-stop-garantie* tot een volgende geschikte stoppositie moeten gelden.

Het is mogelijk bovenstaande te realiseren d.m.v. een “vroege blokkeringszone”. Deze zone schermt geen botsingsarea's af, maar werpt als het ware een schaduw vooruit en zorgt ervoor dat botsingsconfiguraties onbereikbaar worden als voertuigen “normaal” doorrijden. Het configuratiediagram maakt een en ander duidelijk (zie figuur 13).

Bij vroege-blokkering worden potentiële wachttijden niet vergroot, alleen vervroegd. Vroege-blokkeringszones kunnen zoals andere zones met het claim- en release-mechanisme geïmplementeerd worden.

Het lijkt verstandig om naast vroege-blokkeringszones de bestaande afschermingszones voor botsingen te handhaven. Deze zullen evenwel alleen in extreme gevallen (geen normale doorloop van voertuigen) in actie hoeven te komen.

## 7. Zoneuitsluiting

Last-but-not-least is het configuratiediagram een goed hulpmiddel om de cartesische zoneuitsluiting te overzien en eventueel interactief aan te passen. Zoals eerder in dit rapport opgemerkt is verdere ontwikkeling gewenst om de ontwerper te ondersteunen bij het bepalen óf en welke meervoudige, elkaar overlappende cartesische zones in een bepaalde situatie leidt tot verbetering van de doorloop van voertuigen.

## Appendix A Listing road-map-file ECTMODEL.RM

```
NAME ECTMODEL                                RETURN 90
                                             NPOINT 16 16.5, 24.4
SHAPE 2 LENGTH 16 WIDTH 2.5                 POINT 40.0, 24.4
                                             ENDPATH

PATH B11
POINT 0.0,-10.0                             PATH B23
POINT 0.0, 18.5                             POINT 5.0,-10.0
RETURN 90                                    POINT 5.0, 7.3
NPOINT 16 11.5, 30.0                       RETURN 90
POINT 40.0, 30.0                           NPOINT 16 16.5, 18.8
ENDPATH                                    POINT 40.0, 18.8
                                             ENDPATH

PATH B12
POINT 0.0,-10.0                             PATH B24
POINT 0.0, 12.9                             POINT 5.0,-10.0
RETURN 90                                    POINT 5.0, 1.7
NPOINT 16 11.5, 24.4                       RETURN 90
POINT 40.0, 24.4                           NPOINT 16 16.5, 13.2
ENDPATH                                    POINT 40.0, 13.2
                                             ENDPATH

PATH B13
POINT 0.0,-10.0                             PATH B31
POINT 0.0, 7.3                              POINT 9.0,-10.0
RETURN 90                                    POINT 9.0, 18.5
NPOINT 16 11.5, 18.8                       RETURN 90
POINT 40.0, 18.8                           NPOINT 16 20.5, 30.0
ENDPATH                                    POINT 40.0, 30.0
                                             ENDPATH

PATH B14
POINT 0.0,-10.0                             PATH B32
POINT 0.0, 1.7                              POINT 9.0,-10.0
RETURN 90                                    POINT 9.0, 12.9
NPOINT 16 11.5, 13.2                       RETURN 90
POINT 40.0, 13.2                           NPOINT 16 20.5, 24.4
ENDPATH                                    POINT 40.0, 24.4
                                             ENDPATH

PATH B21
POINT 5.0,-10.0                             PATH B33
POINT 5.0, 18.5                             POINT 9.0,-10.0
RETURN 90                                    POINT 9.0, 7.3
NPOINT 16 16.5, 30.0                       RETURN 90
POINT 40.0, 30.0                           NPOINT 16 20.5, 18.8
ENDPATH                                    POINT 40.0, 18.8
                                             ENDPATH

PATH B22
POINT 5.0,-10.0                             PATH B34
POINT 5.0, 12.9                             POINT 9.0,-10.0
```

POINT 9.0, 1.7  
RTURN 90  
NPOINT 16 20.5, 13.2  
POINT 40.0, 13.2  
ENDPATH

PATH B41  
POINT 13.0,-10.0  
POINT 13.0, 18.5  
RTURN 90  
NPOINT 16 24.5, 30.0  
POINT 40.0, 30.0  
ENDPATH

PATH B42  
POINT 13.0,-10.0  
POINT 13.0, 12.9  
RTURN 90  
NPOINT 16 24.5, 24.4  
POINT 40.0, 24.4  
ENDPATH

PATH B43  
POINT 13.0,-10.0  
POINT 13.0, 7.3  
RTURN 90  
NPOINT 16 24.5, 18.8  
POINT 40.0, 18.8  
ENDPATH

PATH B44  
POINT 13.0,-10.0  
POINT 13.0, 1.7  
RTURN 90  
NPOINT 16 24.5, 13.2  
POINT 40.0, 13.2  
ENDPATH

CENTRE 0.2 0.2  
EYE -90 0 10000 10

## Appendix B Listing file ECTMODEL.RRM (klein gedeelte)

Netname: ECTMODEL

Connection Table with 16 entries

Start Node 1 <- Path 4 -> Stop Node 5	Length	58.264
Start Node 1 <- Path 3 -> Stop Node 4	Length	63.864
Start Node 1 <- Path 2 -> Stop Node 3	Length	69.464
Start Node 1 <- Path 1 -> Stop Node 2	Length	75.064
Start Node 6 <- Path 8 -> Stop Node 5	Length	53.264
Start Node 6 <- Path 7 -> Stop Node 4	Length	58.864
Start Node 6 <- Path 6 -> Stop Node 3	Length	64.464
Start Node 6 <- Path 5 -> Stop Node 2	Length	70.064
Start Node 7 <- Path12 -> Stop Node 5	Length	49.264
Start Node 7 <- Path11 -> Stop Node 4	Length	54.864
Start Node 7 <- Path10 -> Stop Node 3	Length	60.464
Start Node 7 <- Path 9 -> Stop Node 2	Length	66.064
Start Node 8 <- Path16 -> Stop Node 5	Length	45.264
Start Node 8 <- Path15 -> Stop Node 4	Length	50.864
Start Node 8 <- Path14 -> Stop Node 3	Length	56.464
Start Node 8 <- Path13 -> Stop Node 2	Length	62.064

### PATH B11

NODE	0.000	1 (	0.000,	-10.000)
CLAIM	0.143	8 (	0.000,	-9.857)
CLAIM	5.743	7 (	0.000,	-4.257)
CLAIM	11.343	6 (	0.000,	1.343)
CLAIM	16.943	5 (	0.000,	6.943)
RELEASE	22.442	8 (	0.000,	12.442)
VIA	28.500	1 (	0.000,	18.500)
RELEASE	29.430	7 (	0.046,	19.428)
VIA	29.629	1 (	0.055,	19.627)
VIA	30.758	1 (	0.221,	20.744)
VIA	31.887	1 (	0.495,	21.838)
VIA	33.016	1 (	0.875,	22.901)
VIA	34.145	1 (	1.358,	23.921)
VIA	35.274	1 (	1.938,	24.889)
VIA	36.403	1 (	2.610,	25.796)
VIA	37.532	1 (	3.368,	26.632)
VIA	38.661	1 (	4.204,	27.390)
CLAIM	38.661	9 (	4.204,	27.390)
VIA	39.790	1 (	5.111,	28.062)
VIA	40.919	1 (	6.079,	28.642)
CLAIM	41.943	10 (	7.005,	29.080)
VIA	42.048	1 (	7.099,	29.125)
VIA	43.177	1 (	8.162,	29.505)
VIA	44.306	1 (	9.256,	29.779)
VIA	45.435	1 (	10.373,	29.945)

VIA 46.564 1 ( 11.500, 30.000)  
RELEASE 46.564 6 ( 11.500, 30.000)  
NODE 75.064 1 ( 40.000, 30.000)  
RELEASE 75.064 10 ( 40.000, 30.000)  
RELEASE 75.064 9 ( 40.000, 30.000)  
RELEASE 75.064 5 ( 40.000, 30.000)  
ENDPATH

PATH B12

NODE 0.000 2 ( 0.000, -10.000)  
CLAIM 0.143 17 ( 0.000, -9.857)  
CLAIM 5.743 16 ( 0.000, -4.257)  
CLAIM 11.343 15 ( 0.000, 1.343)  
CLAIM 16.943 14 ( 0.000, 6.943)  
VIA 22.900 2 ( 0.000, 12.900)  
RELEASE 23.830 17 ( 0.046, 13.828)  
VIA 24.029 2 ( 0.055, 14.027)  
VIA 25.158 2 ( 0.221, 15.144)  
VIA 26.287 2 ( 0.495, 16.238)  
VIA 27.416 2 ( 0.875, 17.301)  
CLAIM 27.485 18 ( 0.905, 17.363)  
VIA 28.545 2 ( 1.358, 18.321)  
VIA 29.674 2 ( 1.938, 19.289)  
VIA 30.803 2 ( 2.610, 20.196)  
VIA 31.932 2 ( 3.368, 21.032)  
CLAIM 32.077 20 ( 3.476, 21.129)  
VIA 33.061 2 ( 4.204, 21.790)  
CLAIM 33.061 19 ( 4.204, 21.790)  
VIA 34.190 2 ( 5.111, 22.462)  
VIA 35.319 2 ( 6.079, 23.042)  
CLAIM 36.343 21 ( 7.005, 23.480)  
VIA 36.448 2 ( 7.099, 23.525)  
VIA 37.577 2 ( 8.162, 23.905)  
VIA 38.706 2 ( 9.256, 24.179)  
VIA 39.835 2 ( 10.373, 24.345)  
VIA 40.964 2 ( 11.500, 24.400)  
RELEASE 40.964 16 ( 11.500, 24.400)  
RELEASE 47.213 14 ( 17.749, 24.400)  
RELEASE 51.213 18 ( 21.749, 24.400)  
RELEASE 55.213 20 ( 25.749, 24.400)  
NODE 69.464 2 ( 40.000, 24.400)  
RELEASE 69.464 21 ( 40.000, 24.400)  
RELEASE 69.464 19 ( 40.000, 24.400)  
RELEASE 69.464 15 ( 40.000, 24.400)  
ENDPATH