

ΠΟΛΥΤΕΧΝΕΙΟ ΚΡΗΤΗΣ
ΤΜΗΜΑ ΗΛΕΚΤΡΟΝΙΚΩΝ ΜΗΧΑΝΙΚΩΝ ΚΑΙ
ΜΗΧΑΝΙΚΩΝ ΥΠΟΛΟΓΙΣΤΩΝ



Διπλωματική Εργασία

Συντονισμένη Στρατηγική Ρομποτικής Ομάδας για το Πρωτάθλημα Τετραπόδων
του RoboCup

Κόντες Γεώργιος

Χανιά 2007

Κατάλογος περιεχομένων

Περίληψη.....	7
1 Εισαγωγή.....	8
2 RoboCup.....	9
2.1 Τι είναι το RoboCup.....	9
2.2 Κατηγορίες.....	9
2.3 Κατηγορία Τετραπόδων (Four Legged League).....	17
2.3.1 SONY AIBO.....	18
2.3.2 Κανονισμοί.....	20
3 Περιγραφή του προβλήματος και σχετική εργασία.....	23
4 Η πρότασή μας	27
4.1 Επιλογή Συστήματος.....	27
4.2 Μεταφορά στο ρομποτικό ποδόσφαιρο.....	28
4.3 Ρόλοι.....	30
4.3.1 Τακτικές άμυνας και επίθεσης.....	30
4.3.2 Συγκεκριμένοι ρόλοι σε κάθε τακτική.....	30
4.3.3 Petri Net Plans.....	32
4.3.4 JARP και PsychoJarp.....	34
4.3.5 Petri Net Plans για τους αντίστοιχους ρόλους.....	36
4.4 Εναλλαγή ρόλων και τακτικών.....	43
5 Σχεδιασμός – Υλοποίηση.....	46
5.1 Επιλογή κώδικα.....	46
5.1.1 Χρησιμοποίηση κώδικα άλλων ομάδων ως βάση και η επιλογή μας.....	46
5.1.2 Περιγραφή και Χαρακτηριστικά του κώδικα.....	47
5.1.3 Δομή κώδικα.....	48
5.1.4 Δημιουργία νέων κλάσεων – ενεργειών.....	49
5.2 Κλάσεις και αντικείμενα σε C++.....	50
5.3 Επικοινωνία.....	53
5.3.1 Προϋπάρχουσα δομή.....	54
5.3.2 Δοκιμές και προβλήματα.....	54
5.3.3 Προτεινόμενη λύση.....	55
5.3.4 Κώδικας C++.....	56
5.3.5 Αποτελέσματα.....	57
6 Συμπεράσματα	60
Βιβλιογραφία.....	62

Κατάλογος εικόνων

Εικόνα 1: 2D Simulation League	10
Εικόνα 2: 3D Simulation League.....	10
Εικόνα 3: Small Size League.....	11
Εικόνα 4: Middle Size League.....	12
Εικόνα 5: Four Legged League.....	13
Εικόνα 6: Humanoid League.....	14
Εικόνα 7: Rescue Robots.....	15
Εικόνα 8: Rescue Simulation.....	15
Εικόνα 9: RoboCup@home.....	16
Εικόνα 10: RoboCup Junior.....	17
Εικόνα 11: Aibo.....	18
Εικόνα 12: AIBO Front.....	19
Εικόνα 13: AIBO Back.....	19
Εικόνα 14: Field (Dimentions in mm) [2].....	22
Εικόνα 15: FSM [2].....	22
Εικόνα 16: The 4-4-2 soccer formation.....	28
Εικόνα 17: The 4-4-2 soccer formation with diamond center.....	28
Εικόνα 18: Offence.....	29
Εικόνα 19: Defence.....	29
Εικόνα 20: Sample Petri Net Plans.....	33
Εικόνα 21: PsychoJarp Petri Net Analyzer.....	34
Εικόνα 22: Counter Attack Attacker.....	36
Εικόνα 23: Counter Attack Midfielder.....	37
Εικόνα 24: Counter Attack Defender.....	37
Εικόνα 25: Passing Attack Attacker.....	38
Εικόνα 26: Passing Attack Midfielder.....	39
Εικόνα 27: Passive Defence Attacker.....	40
Εικόνα 28: Passive Defence Midfielder.....	40
Εικόνα 29: Passive Defence Defender.....	41
Εικόνα 30: Pressing Defence Attacker.....	42
Εικόνα 31: Pressing Defence Midfielder.....	42
Εικόνα 32: Pressing Defence Defender.....	43
Εικόνα 33: Tactic Switching FSM.....	45
Εικόνα 34: modules communication.....	48
Εικόνα 35: field oriented.....	52
Εικόνα 36: Απόδοση του δικτύου πριν τις βελτιώσεις.....	58
Εικόνα 37: Απόδοση του δικτύου μετά τις βελτιώσεις.....	58

Περίληψη

Ο διαγωνισμός RoboCup για πάνω από μια δεκαετία ενισχύει την έρευνα στον τομέα της ρομποτικής, μέσω παιχνιδιών ποδοσφαίρου ανάμεσα σε ομάδες αυτόνομων ρομποτικών πρακτόρων. Η ικανότητα συντονισμού σε μια τέτοια ρομποτική ομάδα είναι ζωτικής σημασίας για την επιτυχία της ομάδας. Σε ανθρώπινες ποδοσφαιρικές ομάδες ο συντονισμός μεταξύ των παικτών επιτυγχάνεται μέσω συστημάτων, τακτικών και στρατηγικών παιχνιδιού, αλλά και μέσω της άμεσης επικοινωνίας των παικτών μεταξύ τους κατά τη διάρκεια του παιχνιδιού. Δυστυχώς η έρευνα στον διαγωνισμό RoboCup έχει επικεντρωθεί κυρίως στη βελτίωση των ατομικών δυνατοτήτων και δεξιοτήτων των παικτών-ρομπότ και σχεδόν καθόλου στην ομαδικότητα και συνεργασία των πρακτόρων μεταξύ τους. Στη διπλωματική εργασία αυτή υιοθετούμε κάποια από τα συστήματα και τις τακτικές που χρησιμοποιούνται σε πραγματικές ομάδες ποδοσφαίρου και τα εφαρμόζουμε στο χώρο του RoboCup. Ορίζουμε διακριτούς ρόλους για κάθε παίκτη και τους υλοποιούμε με τη βοήθεια Petri Net Plans (PNP). Η ανάθεση ενός ρόλου σε κάθε παίκτη κάθε χρονική στιγμή γίνεται δυναμικά χρησιμοποιώντας μια μηχανή πεπερασμένων καταστάσεων (FSM) και ένα απλό και ευέλικτο μοντέλο διεπικοινωνίας μεταξύ των παικτών. Το σύστημα πολυπρακτορικού συντονισμού που προτείνουμε μπορεί, εκτός από το ρομποτικό ποδόσφαιρο να χρησιμοποιηθεί και σε άλλα περιβάλλοντα, όπως η εξερεύνηση πλανητών και αποστολές έρευνας και διάσωσης.

1 Εισαγωγή

Το RoboCup [1] είναι ένας παγκόσμιος διαγωνισμός ρομποτικού ποδοσφαίρου. Στη σύντομη ιστορία του ο διαγωνισμός RoboCup έχει εξελιχθεί σε ένα ετήσιο γεγονός, το οποίο ενώνει μερικούς από τους καλύτερους ερευνητές του τομέα της ρομποτικής από όλο τον κόσμο. Η φύση του διαγωνισμού είναι τέτοια που να απαιτεί αρκετά από τα προβλήματα της ρομποτικής (π.χ αντίληψη, συντονισμό) να επιλύονται σε πραγματικό χρόνο και σε κοινό περιβάλλον για όλες τις διαγωνιζόμενες ομάδες.

Ανάμεσα στις κατηγορίες που περιλαμβάνει είναι και η κατηγορία των τετραπόδων, η οποία μετρά ήδη εννέα χρόνια ιστορίας. Στην κατηγορία αυτή χρησιμοποιούνται τα ρομπότ AIBO της SONY. Κάθε ομάδα αποτελείται από τέσσερις ρομποτικούς παίκτες, οι οποίοι αγωνίζονται σε γήπεδο διαστάσεων 4m x 6m. Κάθε αγώνας περιλαμβάνει δύο ημίχρονα των δέκα λεπτών το καθένα και νικήτρια είναι η ομάδα η οποία έχει σημειώσει τα περισσότερα γκολ στο τέλος του αγώνα. Από τη φύση και μόνο του παιχνιδιού γίνεται αντιληπτό πως είναι απαραίτητη η συνεργασία ανάμεσα στους παίκτες της ομάδας, κάτι που όμως δεν έχει αναπτυχθεί ιδιαίτερα όλα αυτά τα χρόνια ύπαρξης της συγκεκριμένης κατηγορίας. Σκοπός της παρούσας διπλωματικής εργασίας είναι να δώσει μια σημαντική ώθηση προς την κατεύθυνση αυτή, προτείνοντας μια νέα μέθοδο πολυπρακτορικού συντονισμού στο περιβάλλον του ρομποτικού ποδοσφαίρου, με επεκτάσεις και σε άλλα περιβάλλοντα, όπως για παράδειγμα στην πολυπρακτορική πλανητική εξερεύνηση.

Το δεύτερο Κεφάλαιο αποτελεί μια εισαγωγή στον κόσμο του RoboCup, το τρίτο περιγράφει το πρόβλημα που προσπαθούμε να επιλύσουμε και παραθέτει ανάλογες εργασίες από άλλες ομάδες της κατηγορίας και το τέταρτο Κεφάλαιο παρουσιάζει την πρότασή μας για την αποδοτικότερη επίλυση του προβλήματος. Το πέμπτο Κεφάλαιο αναλύει όλα τα στάδια του σχεδιασμού και της υλοποίησης των αλγορίθμων μας, καθώς και τα εργαλεία που χρησιμοποιήσαμε, και το έκτο παραθέτει τα συμπεράσματά μας από την εργασία που εκπονήσαμε, καθώς και μελλοντικές βελτιώσεις της εργασίας αυτής.

2 RoboCup

2.1 Τι είναι το *RoboCup*

Το RoboCup είναι μια παγκόσμια διοργάνωση που προωθεί την έρευνα στην Τεχνητή Νοημοσύνη (TN), τη ρομποτική, καθώς και σε συναφή πεδία. Είναι μια απόπειρα να ενισχυθεί η Τεχνητή Νοημοσύνη και τα ευφυή ρομπότ, παρέχοντας ένα σταθερό πρόβλημα πάνω στο οποίο μια μεγάλη ποικιλία τεχνολογιών μπορούν να υλοποιηθούν και να κριθούν. Η επιτροπή του RoboCup επέλεξε να χρησιμοποιήσει το ποδόσφαιρο σαν κεντρικό θέμα και αντικείμενο έρευνας και μακροπρόθεσμος στόχος είναι το 2050 μια ομάδα από πλήρως αυτόνομα ανθρωποειδή ρομπότ να διαγωνιστεί και να κερδίσει την παγκόσμια πρωταθλήτρια της χρονιάς αυτής στο ποδόσφαιρο [20].

Για να παίξουν τα ρομπότ ποδόσφαιρο, διάφορες τεχνολογίες πρέπει να συνδυαστούν, όπως η πολυπρακτορική συνεργασία και συντονισμός, η μηχανική όραση και η ρομποτική αντίληψη. Εκτός από ποδόσφαιρο, το RoboCup έχει ένα κομμάτι Έρευνας και Διάσωσης (SAR), προωθώντας έτσι την έρευνα σε ευαίσθητους κοινωνικά τομείς της καθημερινής ζωής, καθώς και ένα κομμάτι που προσανατολίζεται σε καθημερινές εργασίες σε οικιακό περιβάλλον.

2.2 Κατηγορίες

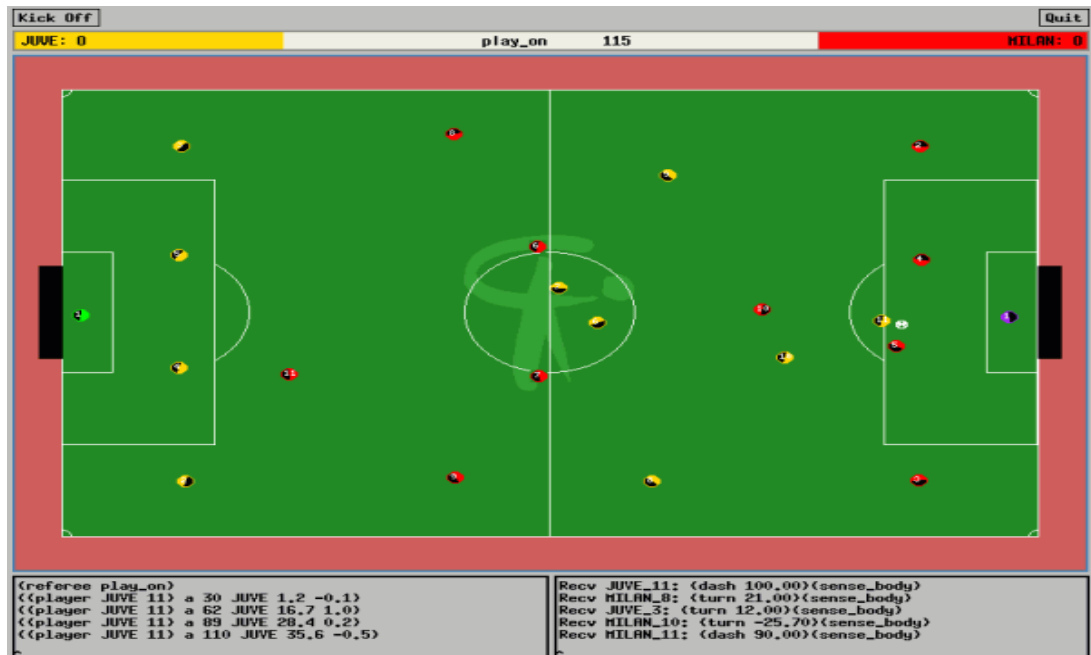
Ο διαγωνισμός RoboCup περιλαμβάνει τις ακόλουθες κατηγορίες [20]:

1. Ρομποτικό Ποδόσφαιρο

- ▢ Κατηγορία εξομοίωσης (Simulation League)

Στην κατηγορία αυτή οι ομάδες προγραμματίζουν έντεκα αυτόνομους

πράκτορες - παίκτες είτε σε δισδιάστατο (2D) είτε σε τρισδιάστατο (3D) περιβάλλον (Εικόνες 1¹, 2²).



Εικόνα 1: 2D Simulation League

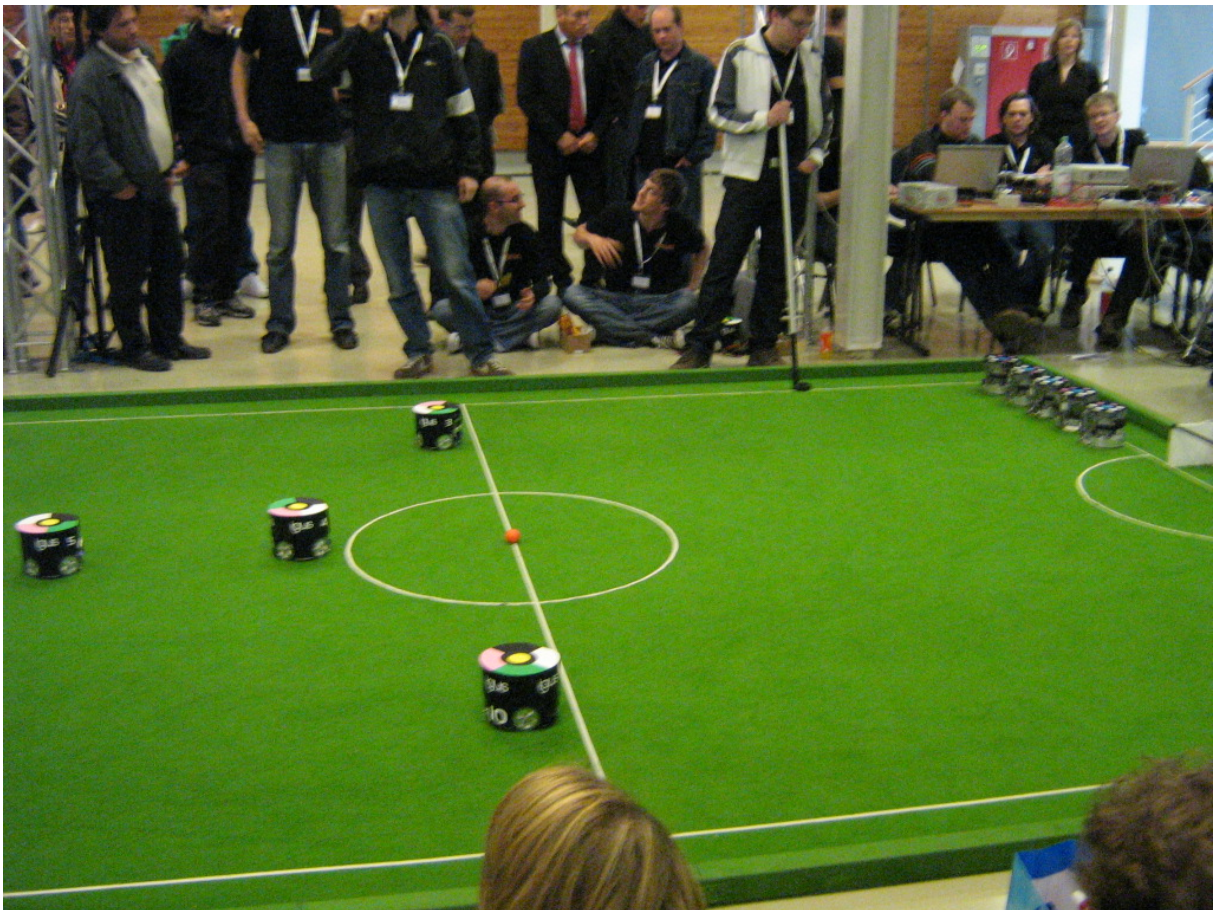


Εικόνα 2: 3D Simulation League

- 1 <http://www.dei.unipd.it/~robocup/old/simulleague/index.html>
- 2 <http://paginas.fe.up.pt/~lpreis/robo2004/Links.htm>

- Κατηγορία Μικρού Μεγέθους (Small Size League)

Στην κατηγορία αυτή οι ομάδες προγραμματίζουν αυτόνομους ρομποτικούς πράκτορες μικρού μεγέθους, οι οποίοι παίζουν ποδόσφαιρο σε γήπεδο διαστάσεων 5m x 3,5m (Εικόνα 3). Κάθε ομάδα αποτελείται από πέντε ρομποτικούς παίκτες.



Εικόνα 3: Small Size League

- Κατηγορία Μεσαίου Μεγέθους (Middle Size League)

Στην κατηγορία αυτή οι ομάδες προγραμματίζουν αυτόνομους ρομποτικούς πράκτορες διαμέτρου περίπου 60cm, οι οποίοι παίζουν ποδόσφαιρο σε γήπεδο διαστάσεων 18m x 12m (Εικόνα 4³). Κάθε ομάδα αποτελείται από έξι παίκτες.



Εικόνα 4: Middle Size League

3 <http://www.robocup2006.org/sixcms/detail.php?id=582>

- Κατηγορία τετραπόδων (Four Legged League)

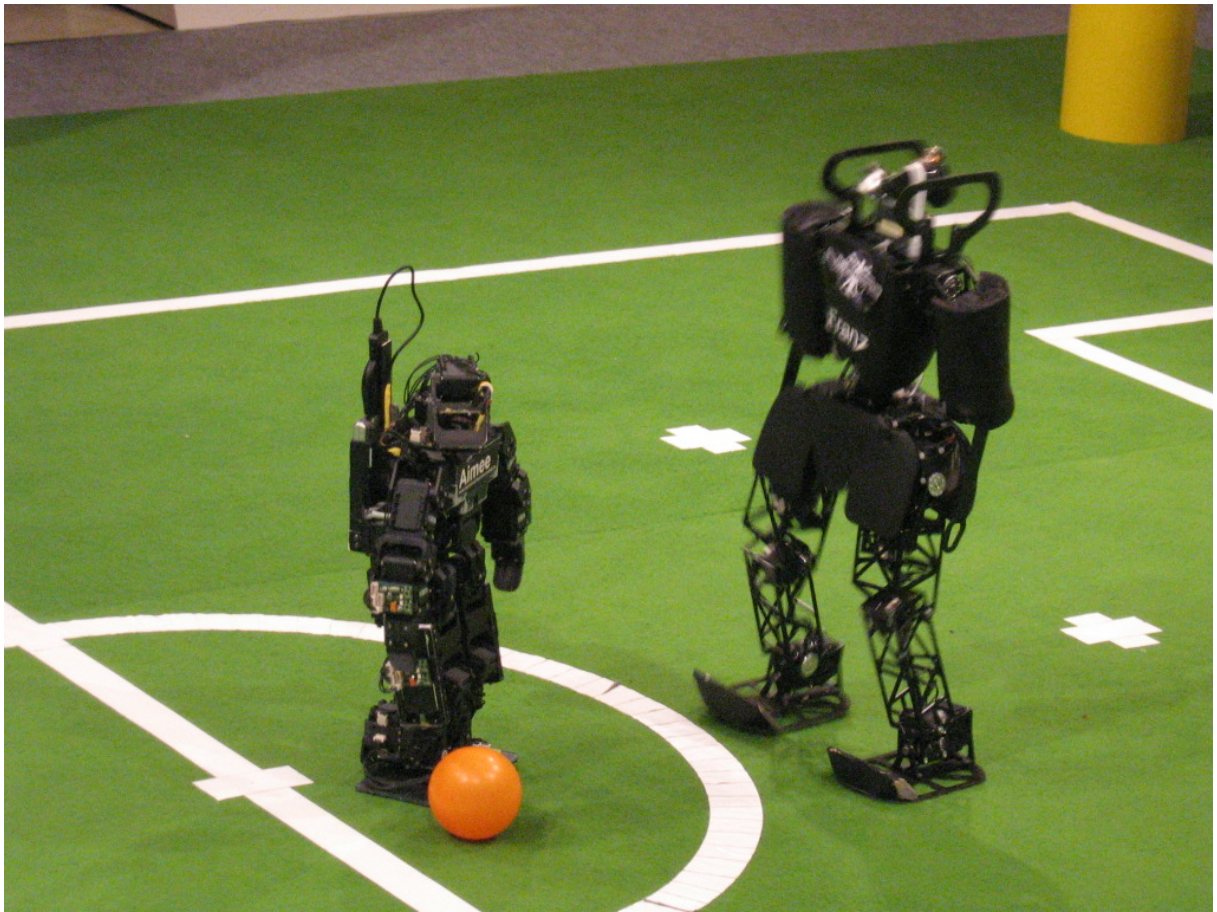
Στην κατηγορία αυτή όλες οι ομάδες συμμετέχουν με τέσσερα ρομπότ AIBO της SONY. Οι αγώνες πραγματοποιούνται σε γήπεδο διαστάσεων 4m x 6m (Εικόνα 5) και κάθε ομάδα αποτελείται από τέσσερα ρομπότ.



Εικόνα 5: Four Legged League

- Κατηγορία ανθρωποειδών (Humanoid League)

Στην κατηγορία αυτή οι ομάδες συμμετέχουν με ρομπότ ιδιοκατασκευές. Κάθε ομάδα αγωνίζεται με δύο ρομπότ σε γήπεδο διαστάσεων 4m x 6m (Εικόνα 6).



Εικόνα 6: Humanoid League

2. Διάσωση

▣ Διασωστικά ρομπότ (Rescue Robots)

Στην κατηγορία αυτή οι ομάδες κατασκευάζουν δικά τους ρομπότ, τα οποία εξερευνούν μια ειδικά κατασκευασμένη πίστα, η οποία προσομοιώνει το τοπίο μετά από μια καταστροφή, για επιζώντες. Τα ρομπότ μπορούν να κινούνται είτε αυτόνομα είτε με τηλεχειρισμό (Εικόνα 7⁴).

▣ Εξομοίωση Διάσωσης (Rescue Simulation)

Στην κατηγορία αυτή οι ομάδες προγραμματίζουν αυτόνομους πράκτορες, ώστε να συμμετέχουν σε αποστολές διάσωσης σε περιβάλλον εξομοίωσης. Η βάση για το περιβάλλον είναι η μηχανή γραφικών του Unreal Tournament 2004 (Εικόνα 8⁵).

4 <http://www.robocup2006.org/sixcms/detail.php?id=576&lang=en>

5 <http://www.science.uva.nl/~arnoud/research/roboresc/virtual2006/>



Εικόνα 7: Rescue Robots



Εικόνα 8: Rescue Simulation

3. RoboCup στο σπίτι

Στην κατηγορία αυτή οι ομάδες προγραμματίζουν ρομπότ, τα οποία έχουν κατασκευάσει οι ίδιες, να εκτελούν διάφορες εργασίες σε οικιακό περιβάλλον (Εικόνα 9).



Εικόνα 9: RoboCup@home

4. Παιδικό RoboCup (RoboCup Junior)

- ▣ Χορός
- ▣ Ποδόσφαιρο
- ▣ Διάσωση

Στο παιδικό RoboCup ισχύουν οι ίδιοι κανόνες όπως και προηγουμένως, αλλά οι διαγωνισμοί γίνονται σε μικρότερα γήπεδα και πίστες, ενώ και τα ρομπότ είναι αρκετά απλούστερα (Εικόνα 10⁶).

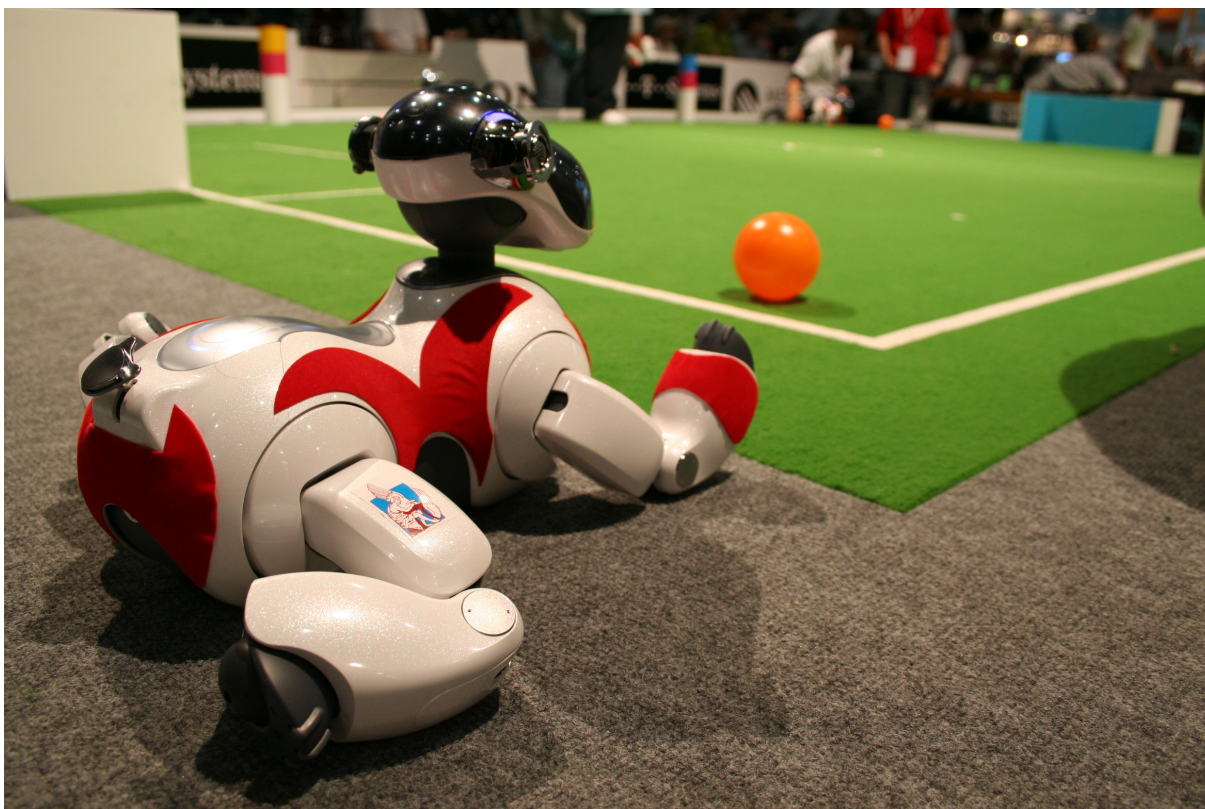


Εικόνα 10: RoboCup Junior

2.3 Κατηγορία Τετραπόδων (Four Legged League)

Στην κατηγορία των τετραπόδων οι ομάδες δεν συμμετέχουν με ρομπότ ιδιοκατασκευές, αλλά με τα τετράποδα ρομπότ AIBO της SONY (Εικόνα 11). Κάθε ομάδα αγωνίζεται με τέσσερα ρομπότ σε ένα γήπεδο διαστάσεων 4m x 6m. Το πρωτάθλημα διέπεται από ορισμένους κανόνες, τους οποίους θα παρουσιάσουμε επιγραμματικά και “περιορίζεται” κατά κάποιον τρόπο από τις δυνατότητες και τα χαρακτηριστικά του ρομπότ που χρησιμοποιείται (Εικόνες 12, 13).

6 <http://www.robocup.org>



Εικόνα 11: Aibo

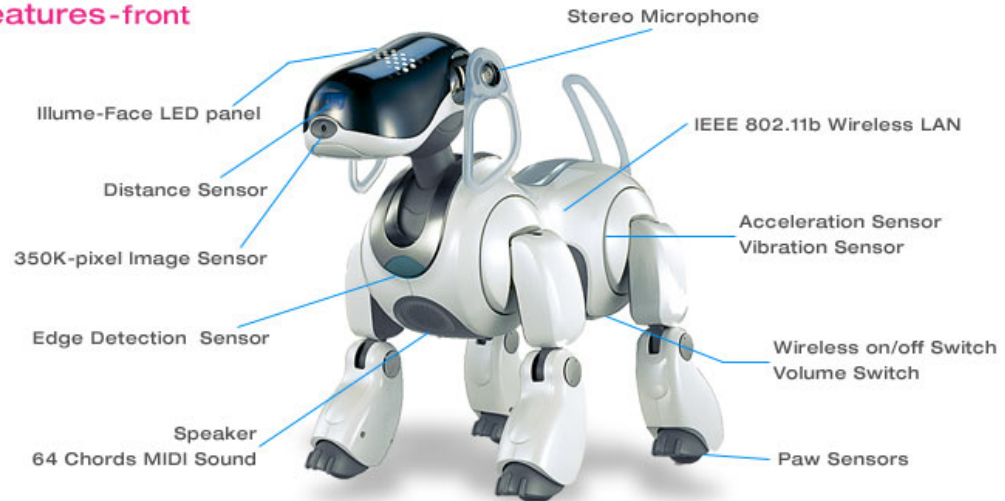
2.3.1 SONY AIBO

Το μοντέλο του ρομπότ AIBO της SONY που χρησιμοποιείται στην κατηγορία των τετραπόδων είναι το ERS – 7. Οι διαστάσεις του είναι 180 x 278 x 319mm και έχει συνολικά είκοσι βαθμούς ελευθερίας. Ο πυρήνας του αποτελείται από έναν 64 – bit RISC επεξεργαστή με συχνότητα ρολογιού στα 576 Mhz και 64MB RAM. Περιέχει μια κάρτα για ασύρματη επικοινωνία στα 2,4GHz και πλήθος πλήκτρων αφής.

Μια πλήρης εικόνα των τεχνικών χαρακτηριστικών του ρομπότ παρατίθεται στις εικόνες 12 και 13⁷. Τα κυριότερα στοιχεία του, τα οποία χρησιμοποιούνται κυρίως στους αγώνες, είναι η κάμερα, οι αισθητήρες υπερύθρων, τα κουμπιά στην πλάτη και στο κεφάλι του και φυσικά το ασύρματο δίκτυο.

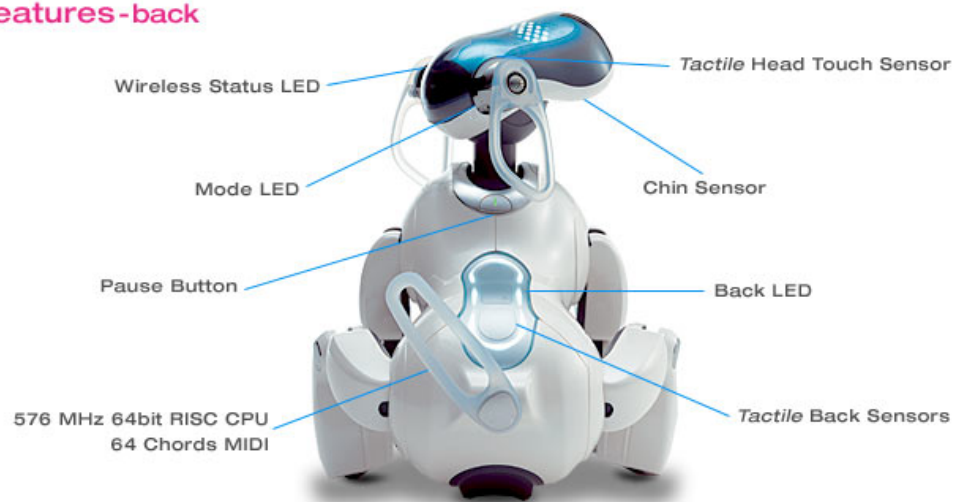
7 http://esupport.sony.com/US/perl/model-documents.pl?mdl=ERS7M3®ion_id=1

► **Features-front**



Εικόνα 12: AIBO Front

► **Features-back**



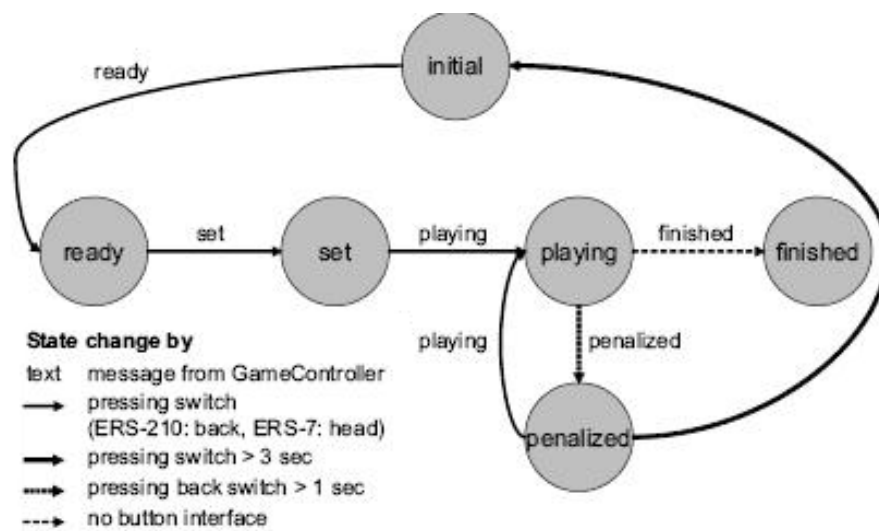
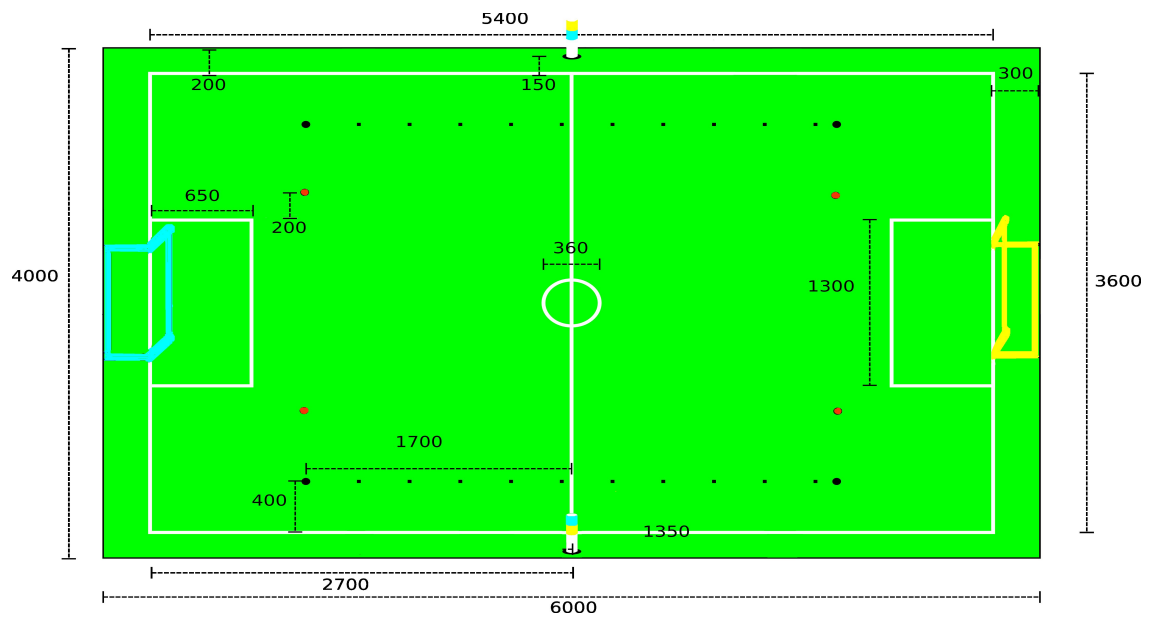
Εικόνα 13: AIBO Back

2.3.2 Κανονισμοί

Οι επίσημοι κανονισμοί [2] του παιχνιδιού είναι εμπνευσμένοι από το πραγματικό ποδόσφαιρο, αλλά με αρκετές διαφοροποιήσεις. Θα προσπαθήσουμε να επισημάνουμε τους κυριότερους από αυτούς και για να αντιληφθούμε το πνεύμα του παιχνιδιού, αλλά και για να γίνει κατανοητή αργότερα η προσέγγισή μας στην τακτική της ομάδας.

1. Οι διαστάσεις του γηπέδου είναι 4m x 6m. Οι διαστάσεις των υπόλοιπων στοιχείων του γηπέδου είναι κι αυτές δεδομένες, όπως φαίνεται και από την Εικόνα 14. Στο γήπεδο τοποθετούνται δύο τέρματα, ένα μπλε κι ένα κίτρινο και στα πλάγια δύο κολώνες με διαφορετικά χρώματα, με τη βοήθεια των οποίων το ρομπότ προσανατολίζεται μέσα στο γήπεδο και εκτιμά την ακριβή θέση του.
2. Ο μέγιστος επιτρεπτός αριθμός ρομπότ για κάθε ομάδα, κάθε στιγμή στο γήπεδο είναι τέσσερα.
3. Ο αγώνας χωρίζεται σε δύο δεκάλεπτα ημίχρονα. Στην ανάπαυλα οι ομάδες αλλάζουν χρώματα και κατεύθυνση επίθεσης.
4. Δεν επιτρέπεται καμία επέμβαση στο υλικό (Hardware) του ρομπότ από την ομάδα.
5. Δεν επιτρέπεται κανενός είδους παρέμβαση στα ρομπότ από μέρους της ομάδας κατά τη διάρκεια του αγώνα, παρά μόνο στην ανάπαυλα.
6. Τα ρομπότ επικοινωνούν με τον GameController (πρόγραμμα σε java για τον έλεγχο του παιχνιδιού), ο οποίος δίνει τις απαραίτητες εντολές κατά τη διάρκεια του παιχνιδιού.
7. Η επικοινωνία γίνεται μόνο μέσω access points που παρέχονται από τους διοργανωτές.
8. Υπάρχουν έξι διαφορετικές καταστάσεις στις οποίες μπορεί να βρεθεί το ρομπότ:
 - Initial: Η αρχική θέση του ρομπότ πριν το παιχνίδι.
 - Ready: Το ρομπότ πρέπει να πάρει την αρχική του θέση μέσα στο γήπεδο.

- ▣ Set: Ο αγώνας είναι έτοιμος να ξεκινήσει και το ρομπότ ψάχνει για την μπάλα.
 - ▣ Play: Το ρομπότ αγωνίζεται.
 - ▣ Penalized: Το ρομπότ δέχεται ποινή εξόδου από το γήπεδο διάρκειας τριάντα δευτερολέπτων για αντικανονική ενέργεια.
 - ▣ Finished: Το παιχνίδι έχει τελειώσει και το ρομπότ παραμένει ακίνητο.
9. Ανάλογα με το ποιο κουμπί πιέζεται και με ποια σειρά, σύμφωνα με τη μηχανή πεπερασμένων καταστάσεων της εικόνας 15 [2] , από τα κουμπιά που βρίσκονται στην πλάτη και το κεφάλι του ρομπότ, αυτό εισέρχεται σε μια από τις παραπάνω καταστάσεις.
10. Οι κυριότερες ενέργειες που τιμωρούνται με απομάκρυνση του ρομπότ από το γήπεδο για τριάντα δευτερόλεπτα είναι:
- ▣ Η κράτηση της μπάλας ανάμεσα στα πόδια και στο κεφάλι για περισσότερα από πέντε δευτερόλεπτα για τον τερματοφύλακα και από τρία δευτερόλεπτα για τους υπόλοιπους παίκτες.
 - ▣ Αν ένα ρομπότ σπρώχνει κάποιο άλλο, ανεξαρτήτου ομάδας, για πάνω από τρία δευτερόλεπτα, εκτός αν είναι τερματοφύλακας και βρίσκεται στην περιοχή του.
 - ▣ Αν κάποιο ρομπότ εγκαταλείψει το χώρο των 4m x 6m.
 - ▣ Αν ο αμυντικός μιας ομάδας εισέλθει στην μικρή περιοχή της ομάδας του.



3 Περιγραφή του προβλήματος και σχετική εργασία

Στο πραγματικό ποδόσφαιρο, από τα σημαντικότερα στοιχεία που πρέπει να έχει μια ομάδα για να κερδίσει είναι η συνοχή και η οργάνωση. Η ύπαρξη των στοιχείων αυτών εξασφαλίζεται μέσω των τακτικών και των συστημάτων που διδάσκονται οι παίκτες κατά τη διάρκεια της προετοιμασίας και των προπονήσεων. Η διατήρηση των στοιχείων αυτών κατά τη διάρκεια ενός αγώνα επιτυγχάνεται μέσω των οδηγιών του προπονητή και κυρίως μέσω της επικοινωνίας μεταξύ των παικτών. Καμιά ομάδα αποτελούμενη από πολύ καλούς παίκτες δεν διακρίθηκε χωρίς να διέπεται από οργάνωση και συνεργασία μεταξύ των παικτών της.

Η οργανωτική επιτροπή της κατηγορίας των τετραπόδων, έχοντας αυτό κατά νου, θέσπισε το 2006 έναν διαγωνισμό πάσας. Τρία ρομπότ τοποθετούνταν σε σχήμα τριγώνου στο γήπεδο και έπρεπε να ανταλλάξουν πάσες μεταξύ τους. Για κάθε επιτυχημένη πάσα κέρδιζαν συγκεκριμένους βαθμούς. Τα αποτελέσματα ήταν αποκαρδιωτικά, ακόμη και για ομάδες που αποτελούν παραδοσιακές δυνάμεις στο χώρο, γεγονός που μας δίνει κάποιες ενδείξεις για τα επίπεδα συνεργασίας των ρομπότ κατά τη διάρκεια των αγώνων.

Για να βγάλουμε ασφαλή και συνολικά συμπεράσματα για το επίπεδο της συνεργασίας μεταξύ των ρομπότ κάθε ομάδας, μπορούμε να ανατρέξουμε στις αναφορές που δημοσιεύει κάθε ομάδα στον δικτυακό τόπο της διοργάνωσης [3]. Παρατηρούμε ότι σχεδόν όλες οι ομάδες ορίζουν διακριτούς ρόλους για κάθε ρομπότ (τερματοφύλακας, επιθετικός, μέσος ή υποστηρικτής και αμυντικός) και κατά τη διάρκεια του τα ρομπότ μπορούν να αλλάξουν ρόλους, ανάλογα με την έκβαση του παιχνιδιού, εκτός από τον τερματοφύλακα που παραμένει σταθερός. Αυτό που διαφέρει είναι η διαδικασία και η μέθοδος με βάση την οποία γίνεται αυτή η αλλαγή. Ακολουθώντας παραθέτουμε τις υλοποιήσεις και τη φιλοσοφία κάποιων αντιπροσωπευτικών ομάδων:

- Η ομάδα ARAIBO έχει δώσει έμφαση στην ανάπτυξη των ατομικών ικανοτήτων κάθε ρομπότ και υλοποιεί ένα απλό μοντέλο εναλλαγής ρόλων [4]: τα ρομπότ υπολογίζουν ατομικά το χρόνο που απαιτείται για το καθένα να

φτάσει στην μπάλα και τον μεταδίδουν μέσω του δικτύου. Το ρομπότ με το μικρότερο χρόνο θα γίνει επιθετικός, με τον αμέσως μικρότερο μέσος και ο ρόλος του εναπομείναντος ρομπότ αποφασίζεται πριν το παιχνίδι, ανάλογα με τη δυναμική του αντιπάλου. Το σημαντικότερο πρόβλημα της προσέγγισης αυτής είναι η απουσία ελέγχου σε περίπτωση αστοχίας δικτύου (κάτι που με βάση την εμπειρία συμβαίνει αρκετά συχνά). Έτσι αν το δίκτυο δεν λειτουργεί όπως θα έπρεπε κάποια στιγμή τα ρομπότ δεν θα λαμβάνουν μηνύματα και δεν θα είναι σε θέση να αποφασίσουν ποια τακτική θα ακολουθήσουν, ή στην καλύτερη περίπτωση θα ακολουθούν την αμέσως προηγούμενη τακτική, με καταστροφικές συνέπειες για την εξέλιξη του παιχνιδιού.

- Παρόμοια στρατηγική, αλλά με σημαντικές βελτιώσεις ακολουθεί η German Team. Κάθε ρομπότ μεταδίδει στο δίκτυο το χρόνο που χρειάζεται να φτάσει στη μπάλα και το ρομπότ με το μικρότερο χρόνο παίρνει το ρόλο του επιθετικού, με τον αμέσως μικρότερο του μέσου και το εναπομείναν ρομπότ του αμυντικού. Η διαφορά με την ομάδα ARAIBO έγκειται στο γεγονός ότι υπάρχει πρόβλεψη για τυχόν αστοχίες δικτύου [5]: αν κάποιο ρομπότ δεν λάβει καινούρια μηνύματα για ορισμένο χρονικό διάστημα, θεωρεί πως το δίκτυο έχει παρουσιάσει κάποιο πρόβλημα και ακολουθεί μια προεπιλεγμένη τακτική. Η τακτική αυτή όμως είναι πολύπλοκη και εξαρτάται άμεσα από τη θέση της μπάλας, με αποτέλεσμα να υπάρχει σύγχυση στην τακτική της ομάδας όταν η θέση της μπάλας δεν είναι διακριτή (π.χ αν η μπάλα βρίσκεται πάνω στη μεσαία γραμμή). Πρέπει να τονιστεί σε αυτό το σημείο πως η German Team προσπάθησε να υλοποιήσει μηχανισμό για πάσες [6], στηριζόμενη και στην όραση και στην επικοινωνία μεταξύ των ρομπότ, όμως διαπιστώθηκε πως η απόδοση ήταν μέγιστη μόνο όταν διακρινόταν ολόκληρο το ρομπότ που θα δεχόταν την πάσα, ενώ η επικοινωνία μέσω του δικτύου ήταν ήδη επιβαρυνμένη και κατά συνέπεια αναξιόπιστη. Παρατηρούμε κι εδώ ότι η χρήση του δικτύου είναι έντονη και η υλοποίηση είναι τέτοια που κάθε χαμένο μήνυμα ισοδυναμεί με προβληματική ανάπτυξη όλη της ομάδας.
- Η ομάδα Cerberus από την Τουρκία ακολουθεί διαφορετική μέθοδο, βασισμένη σε αλγορίθμους δημοπρασιών [7] : κάθε ρομπότ αποστέλλει στο δίκτυο μία και μόνη προσφορά για την εργασία που είναι φτηνότερη γι' αυτό. Οι προσφορές φτάνουν σε όλα τα ρομπότ και κάθε ένα από αυτά αποφασίζει

τοπικά ποια ενέργεια θα ακολουθήσει. Μάλιστα προσπάθησαν να υλοποιήσουν και έναν μηχανισμό για πάσες βασιζόμενοι στην ιδέα αυτή: ο επιθετικός παίρνει προσφορές για πάσες και αν κάποια πάσα είναι συμφέρουσα την εκτελεί, διαφορετικά σουτάρει. Και η συγκεκριμένη υλοποίηση όμως πάσχει στο θέμα της χρήσης του δικτύου. Αν δεν φτάσουν κάποιες προσφορές η συνεργασία της ομάδας δεν είναι αποδοτική.

- Ακόμη μια καλή υλοποίηση προέρχεται από την ομάδα Nubots. Στην υλοποίηση [8] αυτή η στρατηγική είναι σταθερή σε όλη τη διάρκεια του παιχνιδιού και μπορεί να αλλάξει μόνο αν η ομάδα είναι πίσω στο σκορ και ο αγώνας πλησιάζει προς τη λήξη του. Στη συγκεκριμένη υλοποίηση δεν διακρίνονται ρόλοι, αλλά θέσεις (π.χ. κεντρικός αμυντικός, πλάγιος επιθετικός), καθώς και δύο καταστάσεις στις οποίες μπορεί να βρίσκεται ένα ρομπότ: να κυνηγά τη μπάλα ή να βρίσκεται σε κάποια θέση. Και πάλι αποστέλλονται στο δίκτυο οι χρόνοι που απαιτούνται για να φτάσει κάθε ρομπότ στη μπάλα και το ρομπότ με το μικρότερο χρόνο κυνηγά τη μπάλα, ενώ τα υπόλοιπα τοποθετούνται στην πιο κοντινή τους θέση. Αυτή η υλοποίηση, αν και έχει τα ίδια προβλήματα και κινδύνους, όσον αφορά το δίκτυο, όπως και οι παραπάνω, κρύβει και τον κίνδυνο η πιο κοντινή θέση για ένα ρομπότ να είναι για παράδειγμα δεξιά στην άμυνα, ενώ το παιχνίδι εξελίσσεται πολύ μακριά από τη συγκεκριμένη θέση. Αυτό θα έχει σαν αποτέλεσμα το ρομπότ να μπει σε έναν ατέρμονα βρόγχο και να μην μπορεί να εγκαταλείψει τη θέση του, εκτός κι αν οριστεί να κυνηγά τη μπάλα.
- Τέλος υπάρχουν και κάποιες ομάδες, όπως οι Northern Bites [9] που έχουν αναπτύξει μόνο τις ατομικές ικανότητες των ρομπότ και δεν έχουν ασχοληθεί καθόλου με τον τομέα της συνεργασίας.

Με βάση όλα τα παραπάνω διαπιστώνουμε πως στον τομέα της συνεργασίας, της ομαδικότητας και του συστηματικού τρόπου ανάπτυξης του παιχνιδιού οι υλοποιήσεις που υπάρχουν είναι σε πολύ πρώιμο στάδιο, ενώ εναλλαγή πασών δεν συναντάται σε καμία ομάδα.

Τέλος, η όποια συνεργασία υπάρχει σε κάποιες ομάδες, εξαρτάται άμεσα από την καλή λειτουργία του δικτύου και του υλικού των ρομπότ και από την αποδοτικότητα των πρωτοκόλλων επικοινωνίας, αλλά επιβαρύνει σημαντικά την κίνηση στο δίκτυο με μεγάλο όγκο μηνυμάτων, καθ' ένα από τα οποία απαιτείται να καταλήξει σε όλους τους αποδέκτες, για να λειτουργήσει σωστά το σύστημα της ομάδας.

4 Η πρότασή μας

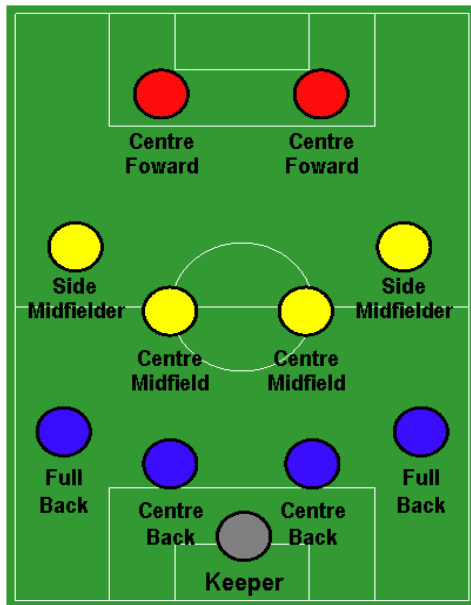
Λαμβάνοντας υπόψιν τις παρατηρήσεις της προηγούμενης παραγράφου γίνεται εύκολα αντιληπτό πως η τακτική σχεδόν όλων των ομάδων είναι αρκετά απλή: υπάρχουν τρεις ρόλοι και αυτοί “διανέμονται” στους παίκτες ανάλογα με το πόσο κοντά βρίσκονται στην μπάλα, χωρίς η “άπληστη” (greedy) αυτή τακτική να εντάσσεται σε κάποιο σύστημα, σχέδιο ή οργανωμένο τρόπο παιχνιδιού.

Η πρόταση μας λοιπόν, και με γνώμονα πως ο απώτερος στόχος του RoboCup είναι ένα παιχνίδι ανάμεσα σε ρομπότ και την παγκόσμια πρωταθλήτρια ποδοσφαίρου, είναι να δανειστούμε και να εφαρμόσουμε σχηματισμούς, συστήματα και πρακτικές δοκιμασμένες για χρόνια σε πραγματικά γήπεδα. Είμαστε πεπεισμένοι πως η πρότασή μας είναι το επόμενο βήμα προς την κατεύθυνση αυτή.

4.1 Επιλογή Συστήματος

Κάθε ομάδα σε έναν πραγματικό αγώνα ποδοσφαίρου αποτελείται από έντεκα παίκτες. Ένας από αυτούς είναι ο τερματοφύλακας και οι υπόλοιποι κινούνται ελεύθερα στο γήπεδο, φυσικά όχι άτακτα, αλλά βάσει κάποιου σχεδίου, κάποιου συστήματος. Ένα από τα παλαιότερα και δημοφιλέστερα συστήματα είναι το 4-4-2. Μια σύγχρονη και επιτυχημένη μετατροπή του είναι το σύστημα 4-4-2 με ρόμβο στο κέντρο (Εικόνες 16, 17)⁸.

⁸[http://en.wikipedia.org/wiki/Formation_\(football\)](http://en.wikipedia.org/wiki/Formation_(football))



Εικόνα 16: The 4-4-2 soccer formation



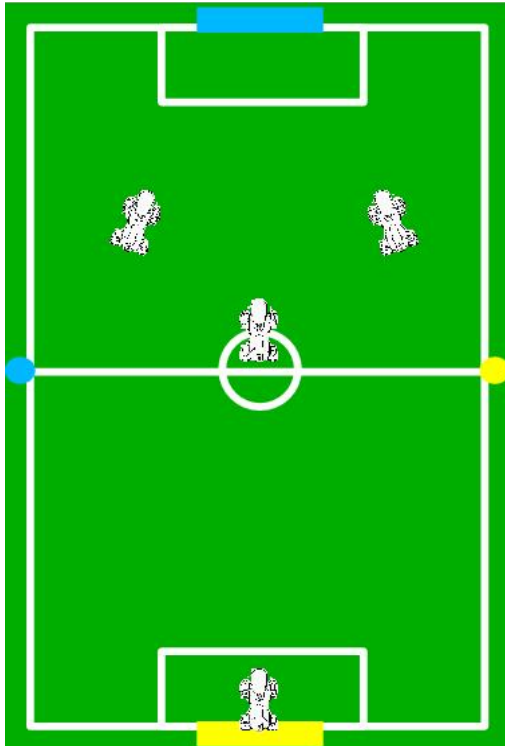
Εικόνα 17: The 4-4-2 soccer formation with diamond center

Το σύστημα αυτό αποτελεί διαχρονική επιλογή της εθνικής ομάδας της Αργεντινής και με το συγκεκριμένο σύστημα η FC Porto κέρδισε το Champions League, υπό τις προπονητικές οδηγίες του Jose Mourinho [10].

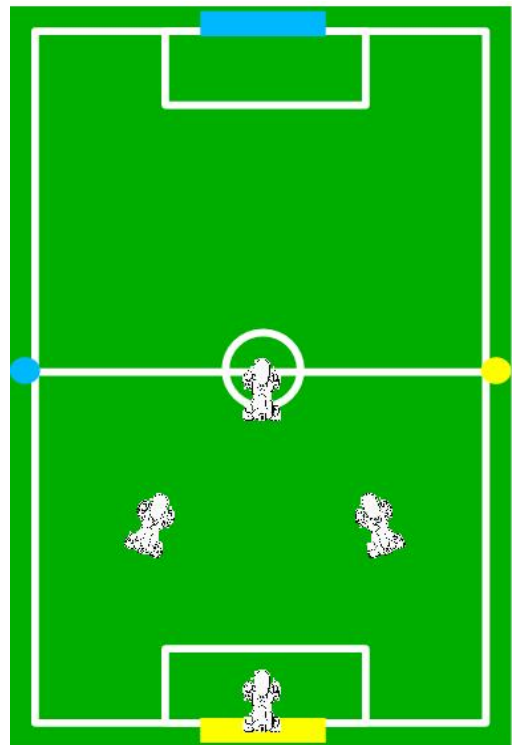
4.2 Μεταφορά στο ρομποτικό ποδόσφαιρο

Για να εφαρμοσθεί το σύστημα που επιλέχθηκε στο ρομποτικό ποδόσφαιρο είναι προφανές πως απαιτούνται κάποιες αλλαγές (μια ομάδα ποδοσφαίρου έχει έντεκα παίκτες και στην κατηγορία των τετραπόδων αγωνίζονται τέσσερα ρομπότ σε κάθε αγώνα). Έτσι λοιπόν, μια και το ένα ρομπότ θα πρέπει να έχει τη θέση του τερματοφύλακα, θεωρήσαμε το γήπεδο του RoboCup σαν μικρογραφία του κέντρου ενός πραγματικού γηπέδου. Τα τρία ρομπότ στο γήπεδο συμπεριφέρονται σαν τους κεντρικούς παίκτες στο πραγματικό σύστημα που επιλέξαμε (οι παίκτες με κίτρινο

χρώμα στην Εικόνα 17) αν αφαιρέσουμε τον έναν. Με τον τρόπο αυτό προκύπτει ένας τριγωνικός σχηματισμός που αλλάζει δυναμικά προσανατολισμό, ανάλογα με τον αν η ομάδα επιτίθεται ή αμύνεται (Εικόνες 18, 19).



Εικόνα 18: Offence



Εικόνα 19: Defence

Παρακολουθώντας έναν σημαντικό αριθμό αγώνων RoboCup παλαιότερων ετών γίνεται αντιληπτό πως όλες οι ομάδες προτιμούν να εκδηλώνουν τις επιθέσεις τους από τον άξονα και όχι από τα άκρα, αφού ο μη σταθερός χειρισμός της μπάλας από τα ρομπότ μπορεί να έχει ως κατάληξη την έξοδο της μπάλας από το γήπεδο. Αντίστοιχα και στην άμυνα, όλες οι ομάδες αμύνονται αποτελεσματικότερα στον άξονα και παραμελούν τα άκρα. Είναι φανερό λοιπόν, παρατηρώντας τους σχηματισμούς που επιλέχθηκαν για την άμυνα και την επίθεση, πως αμυντικά καλύπτεται επαρκώς ο άξονας, ενώ επιθετικά δίνεται η δυνατότητα επίθεσης από τα άκρα.

4.3 Ρόλοι

4.3.1 Τακτικές άμυνας και επίθεσης

Όπως προαναφέρθηκε αποφασίσαμε να εφαρμόσουμε την τριγωνική τακτική στο παιχνίδι μας. Για να το πετύχουμε αυτό διακρίνουμε τέσσερις ρόλους, έναν για κάθε ρομπότ που βρίσκεται στο γήπεδο: τερματοφύλακας (GK), αμυντικός (DEF), μέσος (MID) και επιθετικός (ATT). Εκτός όμως από διαφορετικούς ρόλους κρίνεται αναγκαίο, μια και στόχος μας είναι το οργανωμένο και μεθοδικό παιχνίδι στα πρότυπα του πραγματικού ποδοσφαίρου, να έχουμε και διαφορετικές τακτικές. Έτσι λοιπόν υλοποιούμε δύο τακτικές για επίθεση (την αντεπίθεση ή Counter Attack και την επίθεση με πάσες ή Passing Attack) και δύο τακτικές για άμυνα (την πιεστική άμυνα ή Pressing Defence και την παθητική άμυνα ή Passive Defence). Στο σημείο αυτό πρέπει να τονιστεί πως στην προσέγγισή μας τα ρομπότ δεν ανταλλάσσουν ρόλους κατά τη διάρκεια του παιχνιδιού. Αυτό που μπορεί να αλλάξει είναι η τακτική όλης της ομάδας, η οποία έχει επίδραση και στο ρόλο κάθε παίκτη. Δηλαδή το ρομπότ που θα αναλάβει το ρόλο του επιθετικού από την αρχή, θα παραμείνει επιθετικός καθ' όλη τη διάρκεια του παιχνιδιού. Απλά ο τρόπος που θα παίζει επίθεση θα διαφέρει ανάλογα με την τακτική της ομάδας σε κάθε χρονική στιγμή.

4.3.2 Συγκεκριμένοι ρόλοι σε κάθε τακτική

Στη συνέχεια ακολουθεί η περιγραφή των ρόλων, δηλαδή ο τρόπος παιχνιδιού των ρομπότ σε κάθε τακτική.

1. Αντεπίθεση (Counter Attack)

- ▣ **Επιθετικός:** Ο επιθετικός, έχοντας τη μπάλα στην κατοχή του, κινείται προς το αντίπαλο τέρμα, κατά μέτωπον από το σημείο που κέρδισε τη μπάλα. Όταν φτάσει κοντά στην περιοχή του αντιπάλου είτε σουτάρει απ'ευθείας προς το τέρμα είτε πασάρει στον μέσο, ο οποίος περιμένει στην αντίθετη πλευρά, στο ύψος της γωνίας της περιοχής. Αν εκτελεστεί η πάσα ο

επιθετικός ειδοποιεί τον αμυντικό μέσω του ασύρματου δικτύου.

- ▣ Μέσος: Ενισχύει τον επιθετικό και τοποθετείται στην απέναντι πλευρά της περιοχής από αυτή που βρίσκεται ο επιθετικός περιμένοντας την πάσα. Αν η πάσα είναι επιτυχημένη σουτάρει προς το αντίπαλο τέρμα.
- ▣ Αμυντικός: Ο αμυντικός παραμένει στο κέντρο του γηπέδου, μπροστά από τον κύκλο του κέντρου. Αν η μπάλα πλησιάσει προς το μέρος του, εκτελεί σουτ προς το αντίπαλο τέρμα.

2. Επίθεση με Πάσες (Passing Attack)

- ▣ Μέσος: Ο μέσος κινείται προς την αντίπαλη περιοχή, έχοντας στην κατοχή του τη μπάλα και αναζητώντας μια ευκαιρία να εκτελέσει πάσα προς τον επιθετικό. Τη στιγμή που γίνεται η πάσα, ειδοποιεί τον επιθετικό να υποδεχτεί τη μπάλα, μέσω του ασύρματου δικτύου.
- ▣ Επιθετικός: Ο επιθετικός μετακινείται προς το αντίπαλο τέρμα όντας στραμμένος συνεχώς προς τον μέσο. Καθώς μετακινείται ή όταν φτάσει στην αντίπαλη περιοχή, αναζητά μια πάσα από τον μέσο. Όταν η πάσα πραγματοποιηθεί και είναι επιτυχημένη, η τακτική περνά σε Counter Attack.
- ▣ Αμυντικός: Έχει την ίδια συμπεριφορά με τη συμπεριφορά του Αμυντικού στην τακτική της Αντεπίθεσης που περιγράψαμε νωρίτερα.

3. Πιεστική Άμυνα (Pressing Defence)

- ▣ Επιθετικός: Όσο η μπάλα βρίσκεται στην αντίπαλη πλευρά του γηπέδου, ο επιθετικός κυνηγά τη μπάλα. Όταν η μπάλα εισέλθει στη δική του πλευρά του γηπέδου, τότε κινείται γύρω από τη μεσαία γραμμή, παρακολουθώντας τη μπάλα και τοποθετώντας τον εαυτό του ανάμεσα στη μπάλα και το

αντίπαλο τέρμα.

- ▣ Μέσος: Όσο η μπάλα βρίσκεται στην αντίπαλη πλευρά του γηπέδου, ο μέσος κινείται λίγο μπροστά από τη μεσαία γραμμή και τοποθετείται ανάμεσα στη μπάλα και στο τέρμα της ομάδας του. Μόνο όταν η μπάλα εισέλθει στη δική του πλευρά του γηπέδου την κυνηγά. Οι ενέργειές του είναι συμπληρωματικές του επιθετικού.
- ▣ Αμυντικός: Ο αμυντικός τοποθετείται ανάμεσα στη μπάλα και στο τέρμα της ομάδας του, με τον περιορισμό ότι δεν περνά ποτέ τη γραμμή του κέντρου.

4. Παθητική Άμυνα (Passive Defence)

- ▣ Επιθετικός: Όσο η μπάλα βρίσκεται στην αντίπαλη πλευρά του γηπέδου, ο επιθετικός κινείται πάνω στη μεσαία γραμμή και τοποθετείται ανάμεσα στη μπάλα και στο τέρμα της ομάδας του. Όταν η μπάλα εισέλθει στη δική του πλευρά του γηπέδου η τακτική γίνεται Pressing Defence.
- ▣ Μέσος: Ο μέσος υποστηρίζει τον επιθετικό, μένοντας πίσω του και παρατηρώντας τη μπάλα κάθε στιγμή.
- ▣ Αμυντικός: Ο αμυντικός τοποθετείται ανάμεσα στη μπάλα και στο τέρμα της ομάδας του, ακριβώς μπροστά από την περιοχή του τέρματός του.

4.3.3 Petri Net Plans

Οι ρόλοι που προαναφέραμε υλοποιήθηκαν με τη βοήθεια των Petri Net Plans (PNP) [11]. Η δομή, η μορφή και οι ιδιότητες των Petri Net Plans επιτρέπουν τη χρήση περίπλοκων δομών και ενεργειών, όπως συγχρονισμό μεταξύ των ενεργειών

του ρομπότ και διακοπών (interrupts) των ενεργειών.

Ένα Petri Net [12],[13] είναι ένας τύπος γραφικής γλώσσας, η οποία χρησιμοποιείται για την μοντελοποίηση δυναμικών συστημάτων. Επιτρέπει την περιγραφή των συστημάτων αυτών με τη χρήση κατευθυνόμενων γράφων με βάρη, όπου οι κόμβοι δηλώνουν καταστάσεις (κύκλοι) ή μεταβάσεις (ορθογώνια) και οι ακμές ανάμεσά τους αντιπροσωπεύουν πιθανά μονοπάτια εκτέλεσης, αλλά και συγκεκριμένα βάρη. Η ροή της εκτέλεσης δηλώνεται εισάγοντας την έννοια της μάρκας (token). Οι μάρκες διακινούνται ανάμεσα στις καταστάσεις προκαλώντας στις μεταβάσεις να ενεργοποιούνται, αν ικανοποιούνται όλες οι προϋποθέσεις. Μια παραλλαγή των Petri Nets, τα Extended Petri Nets[13] επιτρέπουν την ύπαρξη παρεμποδιστικών ακμών, των οποίων η συμπεριφορά ενεργοποίησης λειτουργεί συμπληρωματικά ως προς αυτή των κανονικών ακμών.

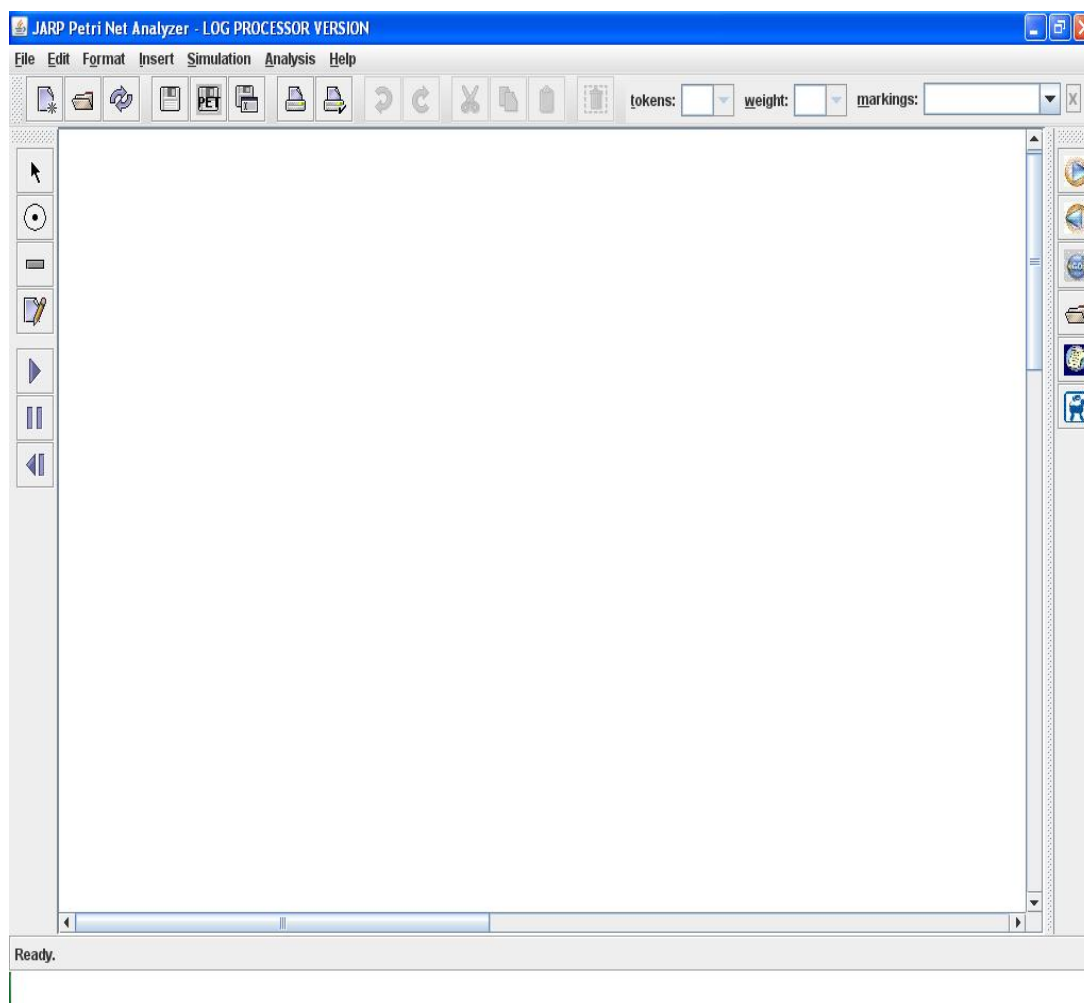
Ένα Petri Net Plan είναι μια συλλογή από ενέργειες δομημένες σαν ένα Extended Petri Net με ακμές χωρίς βάρη. Κάθε ενέργεια στο πλάνο περιγράφεται ακριβώς και αποκλειστικά από τρεις φάσεις (αρχικοποίηση ή start, εκτέλεση ή execute, τέλος ή end). Οι φάσεις τις αρχικοποίησης και του τέλους αναπαριστώνται ως μεταβάσεις (ορθογώνια) και η φάση της εκτέλεσης αναπαριστάται ως μια κατάσταση (κύκλος). Εκτός όμως από μεμονωμένες ενέργειες, σε ένα PNP μπορούν να καλούνται και να εκτελούνται άλλα PNP που έχουν οριστεί αλλού. Ένα παράδειγμα των δομών αυτών φαίνεται στην Εικόνα 20. Στη συνέχεια της εργασίας αυτής, το όνομα όποιας φάσης του PNP (αρχικοποίηση, εκτέλεση, τέλος) έχει το πρόθεμα “Act”, θα θεωρείται ότι αναφέρεται σε μια μεμονωμένη ενέργεια, διαφορετικά θα αποτελεί κλήση σε κάποιο άλλο PNP, που πιθανότατα περιέχει πολλές ενέργειες.



Εικόνα 20: Sample Petri Net Plans

4.3.4 JARP και PsychoJarp

Για τη δημιουργία και γραφική απεικόνιση των Petri Net Plans χρησιμοποιείται το εργαλείο JARP [14]. Για να μπορέσουμε όμως να συνδέσουμε τη γραφική απεικόνιση των ενεργειών με τις κλάσεις και τα αντικείμενα των κλάσεων που έχουν δηλωθεί, αλλά και για να τις μετατρέψουμε σε μορφή που να μπορεί να επεξεργαστεί το ρομπότ, χρησιμοποιούμε το εργαλείο PsychoJarp (Εικόνα 21), το οποίο έχει ακριβώς το ίδιο interface με το Jarp και απλά κάποιες πρόσθετες λειτουργίες. Αφού κατασκευάσουμε το κατάλληλο διάγραμμα με το PsychoJarp πατάμε το κουμπί “pet” και το πρόγραμμα παράγει ένα αρχείο με κατάληξη “pet” και με το ίδιο όνομα με το όνομα που έχουμε δώσει στο Petri Net που κατασκευάσαμε. Ένα ενδεικτικό “.pet” αρχείο φαίνεται στην πιο κάτω σελίδα. Το αρχείο αυτό είναι σε τέτοια μορφή, ώστε να μπορεί να επεξεργαστεί από το ρομπότ.



Εικόνα 21: PsychoJarp Petri Net Analyzer

#keeperSideWalk.pet#

NAME

keeperSideWalk

GOALSTATE

0 0 0 0 0 1

PLACESINFO

0	normal					
0	normal					
1	normal					
0	exec	ActKeeperSideWalkingVsBall				2
0	exec	trackBallKeeper	3			
0	normal					

TRANSITIONSINFO

simple	
start	trackBallKeeper
end	ActKeeperSideWalkingVsBall
end	trackBallKeeper
start	ActKeeperSideWalkingVsBall

TANSPINCMATRIX

1 0 0 0 -1
1 -1 0 0 0
-1 0 0 0 0
0 0 -1 0 1
0 1 -1 -1 0
0 0 1 0 0

MINUSINCMATRIX

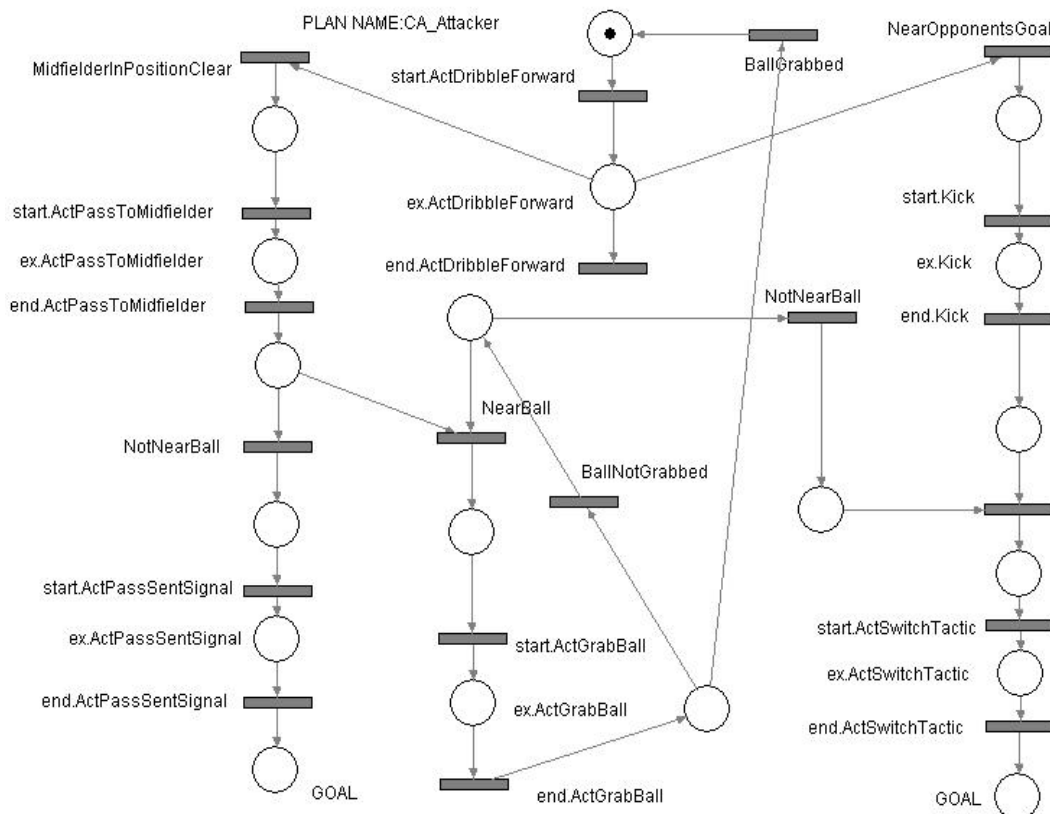
0 0 1 0 0 0
0 1 0 0 0 0
0 0 0 1 1 0
0 0 0 0 1 0
1 0 0 0 0 0

4.3.5 Petri Net Plans για τους αντίστοιχους ρόλους

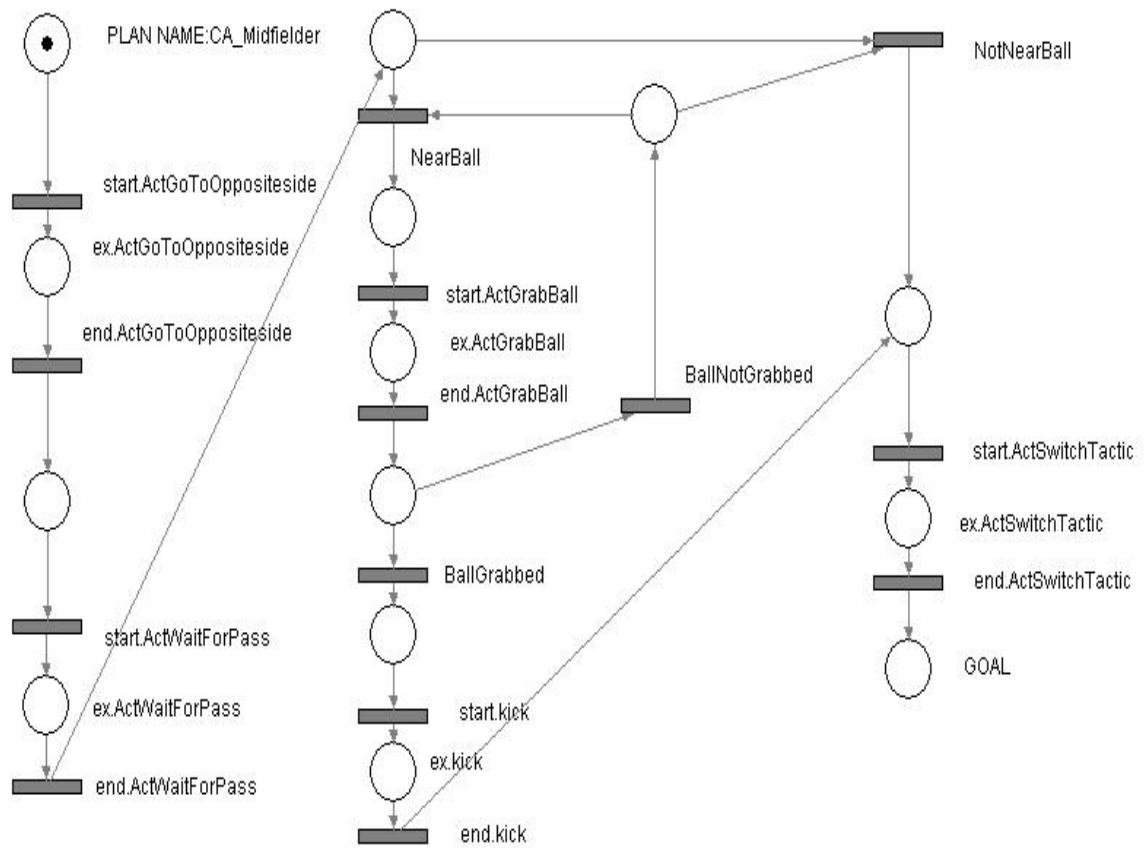
Στη συνέχεια ακολουθούν οι ρόλοι σε κάθε τακτική με τη μορφή PNP. Πρέπει να τονιστεί πως τα ονόματα των καταστάσεων των PNP δεν είναι τα πραγματικά ονόματα των αντικειμένων των κλάσεων που υλοποιήθηκαν, αλλά είναι αντιπροσωπευτικά και περιγραφικά των ενεργειών που καλούν και αντιπροσωπεύουν.

1. Αντεπίθεση

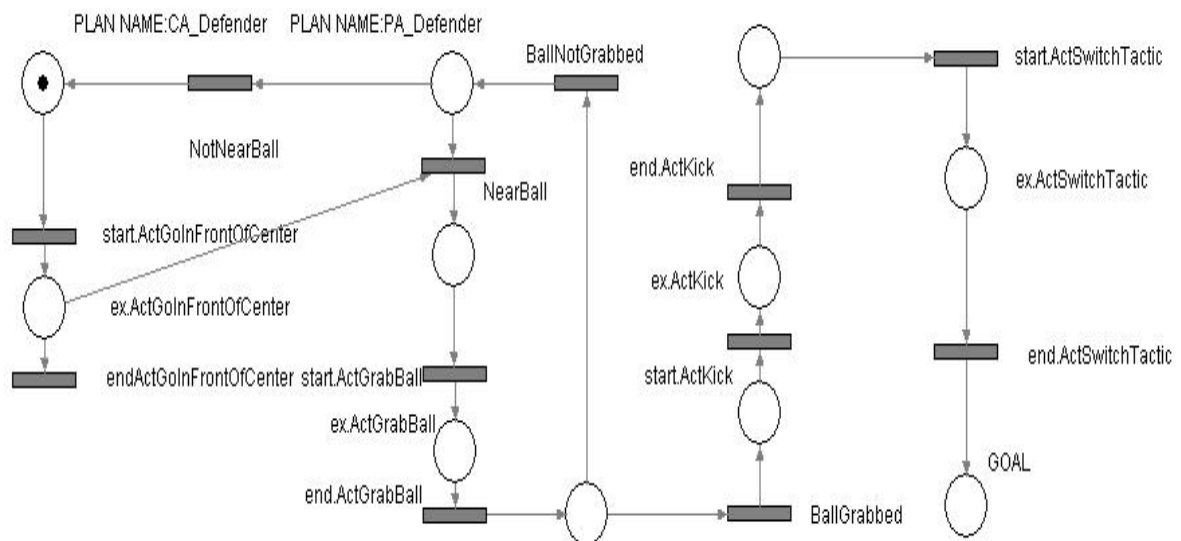
- Επιθετικός (*Εικόνα 22*)
- Μέσος (*Εικόνα 23*)
- Αμυντικός (*Εικόνα 24*)



Εικόνα 22: Counter Attack Attacker



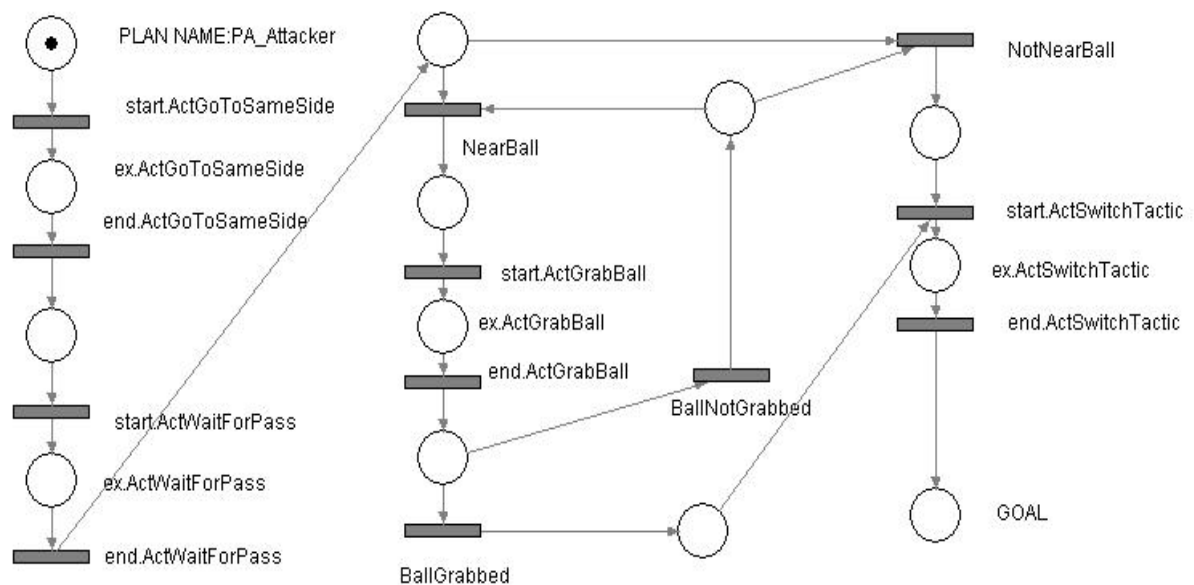
Εικόνα 23: Counter Attack Midfielder



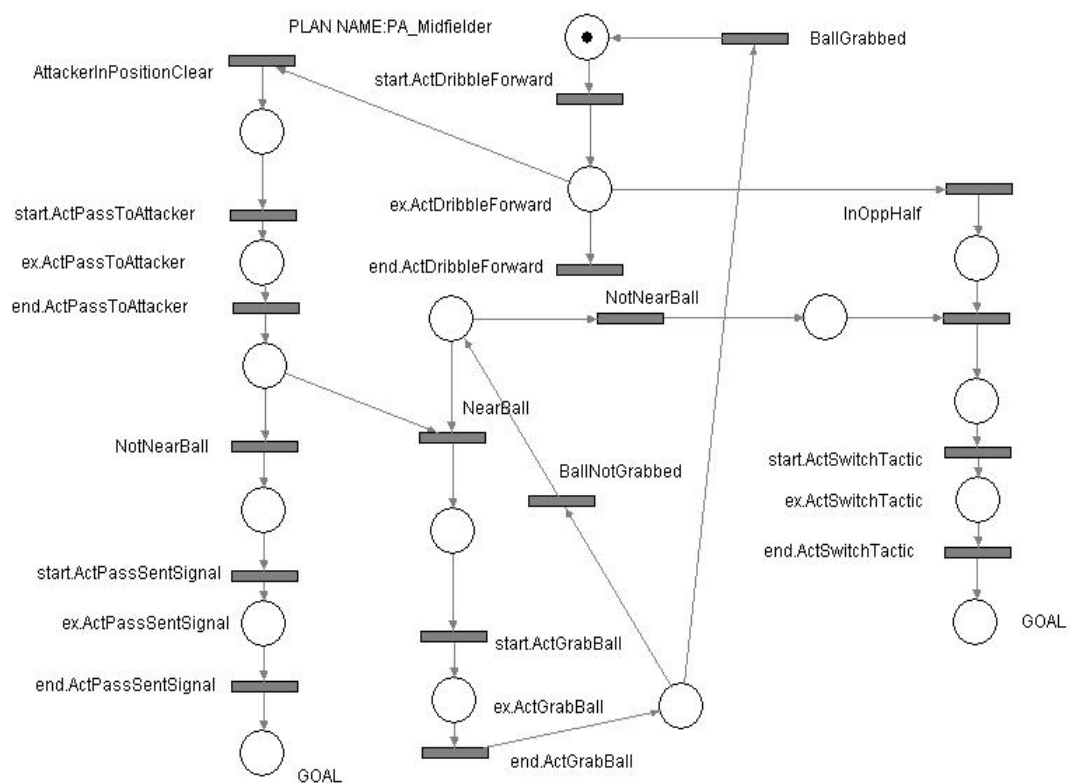
Εικόνα 24: Counter Attack Defender

2. Επίθεση με πάσες

- Επιθετικός (*Εικόνα 25*)
- Μέσος (*Εικόνα 26*)
- Αμυντικός (*ίδιο PNP με τον αμυντικό στην Αντεπίθεση*)



Eikóna 25: Passing Attack Attacker



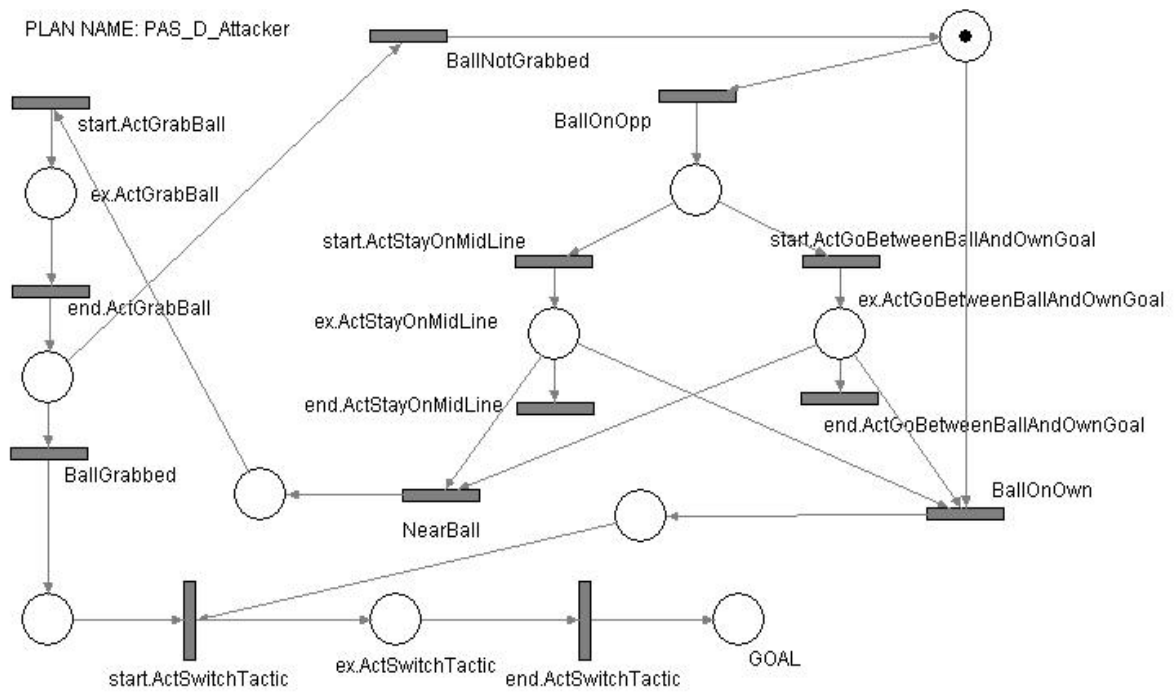
Εικόνα 26: Passing Attack Midfielder

3. Παθητική Άμυνα

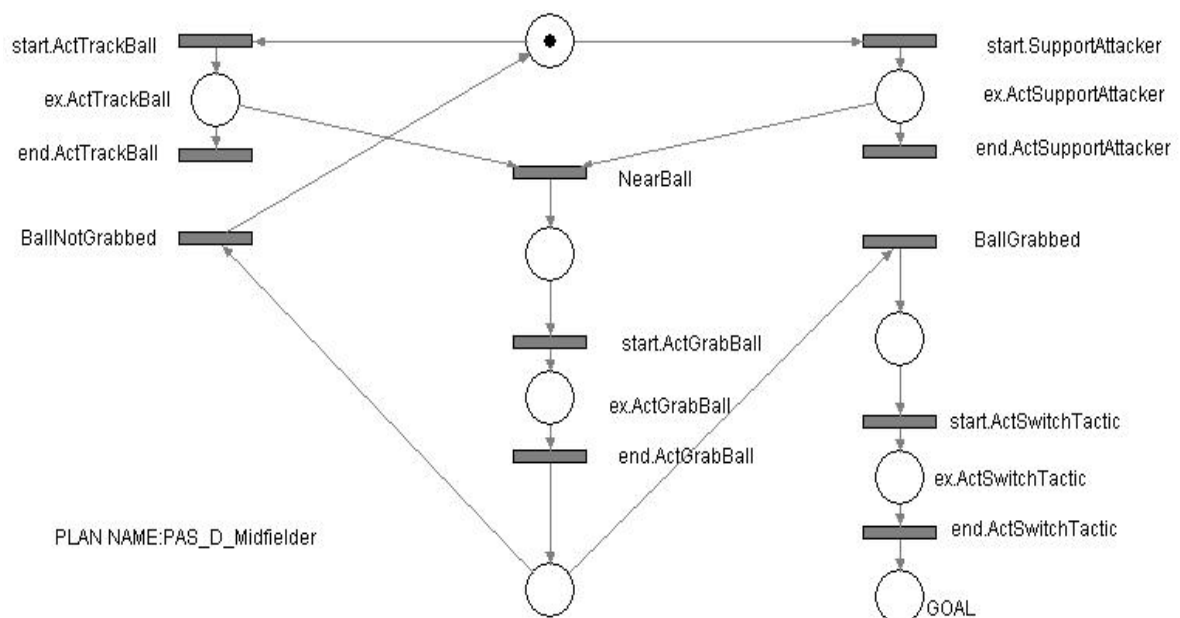
□ Επιθετικός (Εικόνα 27)

□ Μέσος (Εικόνα 28)

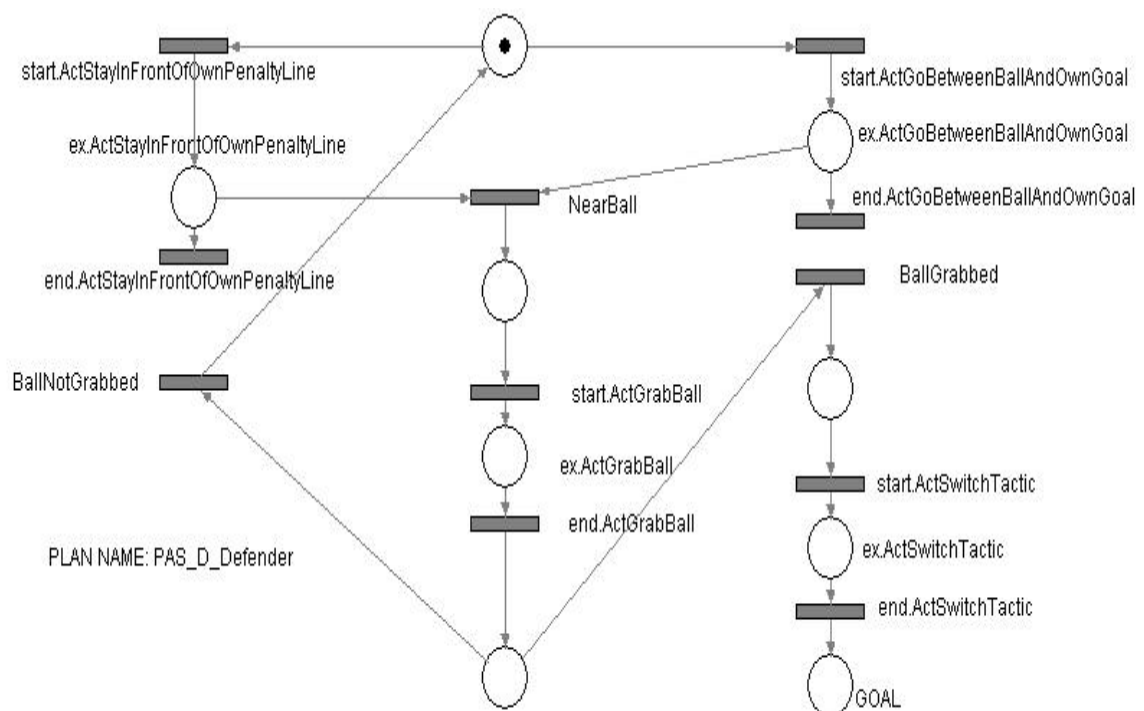
□ Αμυντικός (Εικόνα 29)



Εικόνα 27: Passive Defence Attacker



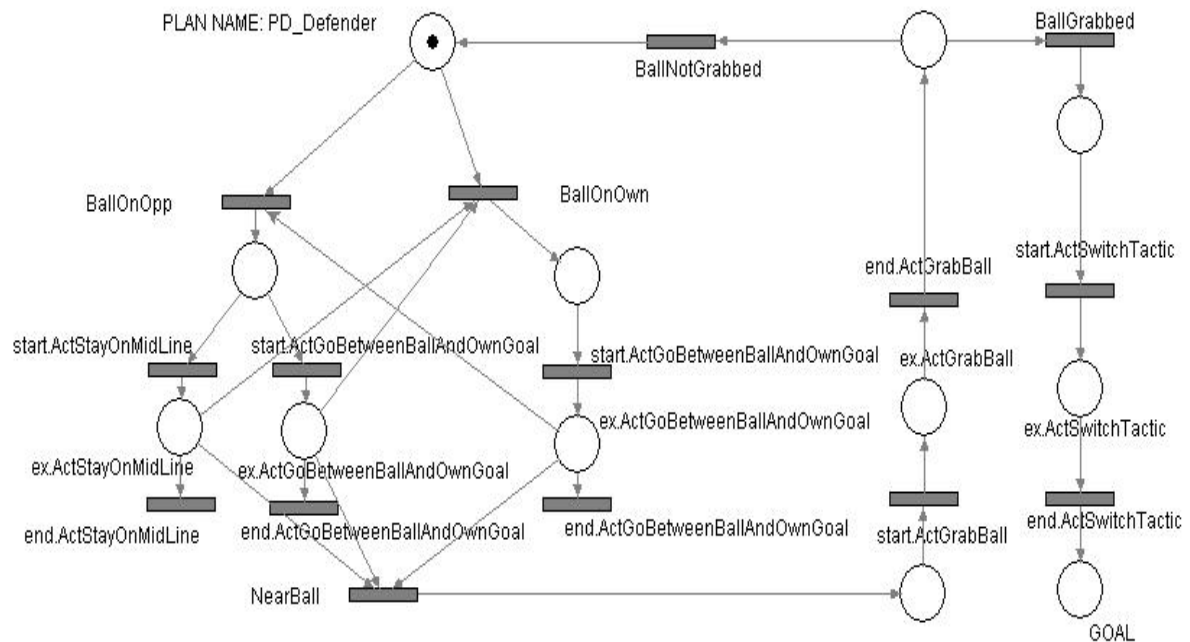
Εικόνα 28: Passive Defence Midfielder



Εικόνα 29: Passive Defence Defender

4. Πιστική Άμυνα

- Επιθετικός (Εικόνα 30)
- Μέσος (Εικόνα 31)
- Αμυντικός (Εικόνα 32)



Εικόνα 32: Pressing Defence Defender

Συμπληρωματικά πρέπει να αναφερθεί πως για λόγους απλότητας και καλύτερης κατανόησης των σχημάτων δεν συμπεριλαμβάνονται κάποιες απλές και απαραίτητες, αλλά ευκόλως εννοούμενες ενέργειες, όπως για παράδειγμα η αναζήτηση της μπάλας αν αυτή δεν βρίσκεται στο οπτικό πεδίο του παίκτη.

4.4 Εναλλαγή ρόλων και τακτικών

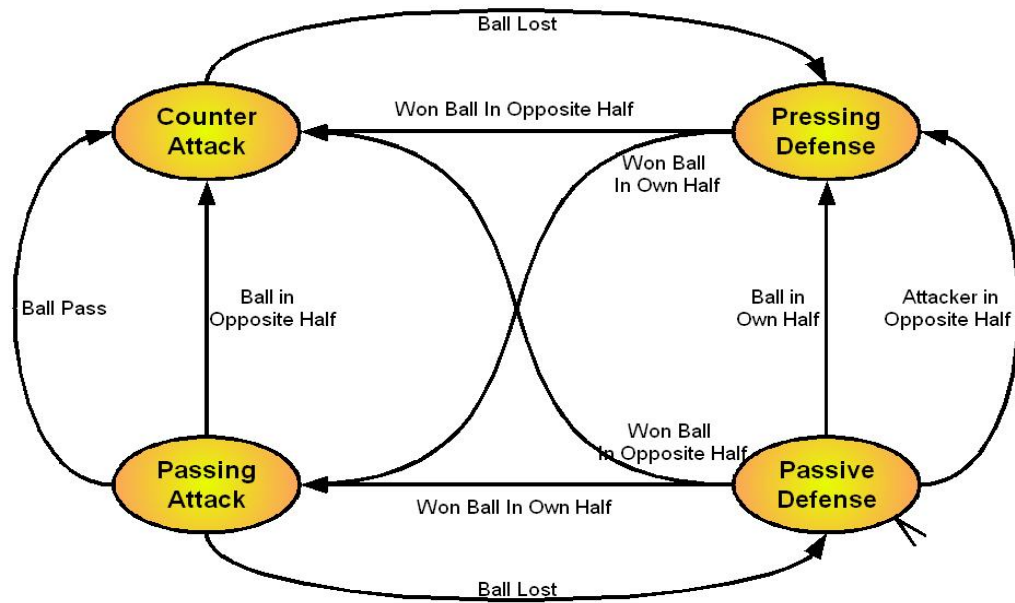
Για να είναι οργανωμένο το παιχνίδι της ομάδας δεν αρκεί να ορίσουμε τους ρόλους που μπορεί να έχει ο κάθε παίκτης, αλλά και να μπορούν οι παίκτες να εναλλάσσονται στους επιμέρους ρόλους, ανάλογα με τα δεδομένα του παιχνιδιού κάθε στιγμή, λαμβάνοντας υπόψιν τα επιμέρους προβλήματα που μπορεί να ανακύψουν, όπως αναφέρθηκαν στην ενότητα 3. Είναι απαραίτητο λοιπόν να σχεδιαστεί ένας μηχανισμός εναλλαγής ρόλων των παικτών που να μην απαιτεί εκτεταμένη χρήση του ασύρματου δικτύου, που να προσαρμόζεται σε τυχόν αστοχίες του δικτύου και που να είναι αποδοτικός “ποδοσφαιρικά”, δηλαδή να επιτρέπει στους

παίκτες να αγωνίζονται οργανωμένα, ακολουθώντας τις τακτικές που αναλύσαμε προηγουμένως.

Για τους λόγους αυτούς, επιλέχθηκε η μη εναλλαγή ρόλων, αλλά η εναλλαγή των τακτικών κατά τη διάρκεια του παιχνιδιού. Αντί τα ρομπότ να στέλνουν μέσω του δικτύου τους χρόνους που χρειάζονται για να προσεγγίσουν τη μπάλα ή την προσφορά τους, αν υιοθετηθεί η μέθοδος των δημοπρασιών, αποστέλλεται μόνο ένα σήμα από ένα ρομπότ. Το σήμα αυτό περιέχει μόνο πληροφορία για την ταυτότητα του αποστολέα, αν κέρδισε τη μπάλα ή την έχασε και την τρέχουσα θέση του αν είναι επιθετικός. Αυτό είναι και το μόνο μήνυμα στο δίκτυο. Γίνεται εύκολα αντιληπτό πως η κίνηση στο δίκτυο μειώνεται σημαντικά με αυτό τον τρόπο, αφού δεν αποστέλλονται μηνύματα από όλα τα ρομπότ, τα οποία μηνύματα είναι απαραίτητο να φτάσουν σε όλα τα υπόλοιπα ρομπότ.

Στη συνέχεια τα ρομπότ αποφασίζουν τοπικά ποια τακτική θα ακολουθήσουν με μόνα δεδομένα τη θέση του επιθετικού στο γήπεδο και τη θέση της μπάλας, σύμφωνα με τη μηχανή πεπερασμένων καταστάσεων της εικόνας 33. Αν τα ρομπότ δεν λάβουν νέο μήνυμα για συγκεκριμένο χρονικό διάστημα, υποθέτουν βλάβη στο δίκτυο και υιοθετούν αυτόματα τη συντηρητικότερη τακτική, την Παθητική Άμυνα. Πρέπει να τονίσουμε πως δεν υπάρχει ανάγκη για εναλλαγή ρόλων με την προσέγγισή μας. Κάθε ρομπότ διατηρεί το ρόλο που του ανατέθηκε αρχικά καθ' όλη τη διάρκεια του παιχνιδιού, απλά τα επιμέρους χαρακτηριστικά του κάθε ρόλου διαφοροποιούνται, ανάλογα με την τακτική που ακολουθείται τη δεδομένη χρονική στιγμή.

Τα πλεονεκτήματα της μεθόδου αυτής είναι η μη καταπόνηση, και επομένως αποδοτικότερη χρήση του δικτύου, η εναλλακτική λύση σε περίπτωση βλάβης του δικτύου ή του υλικού του κάθε ρομπότ μεμονωμένα και η μη ύπαρξη ενός μόνο ρομπότ που να παίρνει τις αποφάσεις και να τις ανακοινώνει στα υπόλοιπα, με καταστροφικές συνέπειες αν το ρομπότ αυτό αποβληθεί, τεθεί εκτός λειτουργίας ή έχει λάθος δεδομένα για το παιχνίδι. Κάθε ρομπότ αποφασίζει τοπικά και με βάση τα δεδομένα που έχει το ίδιο για το ποια τακτική θα ακολουθήσει, έτσι ώστε ακόμη κι αν πάρει τη λάθος απόφαση, τα υπόλοιπα ρομπότ να συνεχίσουν να ανταποκρίνονται στις ανάγκες του παιχνιδιού.



Εικόνα 33: Tactic Switching FSM

5 Σχεδιασμός – Υλοποίηση

5.1 Επιλογή κώδικα

5.1.1 Χρησιμοποίηση κώδικα άλλων ομάδων ως βάση και η επιλογή μας

Στην κοινότητα των ομάδων που συμμετέχουν στο RoboCup, και ειδικά στην κατηγορία των τετραπόδων που έχει αρκετά χρόνια ιστορίας και ο τύπος των ρομπότ είναι κοινός για όλες τις ομάδες, είναι συνηθισμένη και απόλυτα αποδεκτή πρακτική οι νέες ομάδες να ξεκινούν βασιζόμενες σε κώδικα που δημοσιεύουν παλιότερες και πιο έμπειρες ομάδες. Αυτό επιτρέπεται επειδή η λογική όλου του εγχειρήματος, της διοργάνωσης αυτής, είναι η πρόοδος στους τομείς της επιστήμης που προαναφέραμε, χωρίς να υπάρχει στεία ανταγωνιστικότητα ανάμεσα στις ομάδες. Παρέχοντας οι παλιότερες ομάδες τον κώδικά τους σε άλλες ομάδες, εξασφαλίζεται ότι οι νέες ομάδες δεν θα αναλωθούν ή θα αποθαρρυνθούν από την πολλή δουλειά που απαιτείται για να υλοποιηθούν απλές λειτουργίες, απαραίτητες όμως σε μια ομάδα (π.χ περπάτημα, επικοινωνία των διεργασιών μεταξύ τους, επικοινωνία των ρομπότ μέσω του δικτύου κ.ά.), αλλά θα ξεκινήσουν να προγραμματίζουν τα ρομπότ και να υλοποιούν ιδέες και αλγορίθμους σε υψηλότερο επίπεδο, ώστε τελικά η αναμέτρηση να μεταφερθεί ουσιαστικά στο επίπεδο αυτό.

Φυσικά υπάρχει και η αρνητική πλευρά αυτής της παράδοσης, που δεν είναι άλλη από τους περιορισμούς που ανακύπτουν από τη χρήση συγκεκριμένης υλοποίησης. Ακόμη κι αν είναι δυνατόν να επαναπρογραμματιστεί οποιαδήποτε συμπεριφορά ή ικανότητα του ρομπότ κριθεί μη επαρκής ή μη σύμφωνη με τα σχέδια της ομάδας (π.χ. αν το υπάρχον περπάτημα θεωρηθεί αργό, προγραμματίζεται ένα καινούριο σύμφωνα με τα δεδομένα της ομάδας, ή αν τα πρωτόκολλα επικοινωνίας μεταξύ των ρομπότ δεν ικανοποιούν, μπορούν να υλοποιηθούν διαφορετικά), ωστόσο αυτό που δεν μπορεί να αποφευχθεί είναι πως πρέπει να ακολουθηθεί η ίδια οργάνωση κώδικα, πράγμα που μπορεί να στερεί σε ευελιξία σε αρκετές περιπτώσεις, όπως θα φανεί στη συνέχεια.

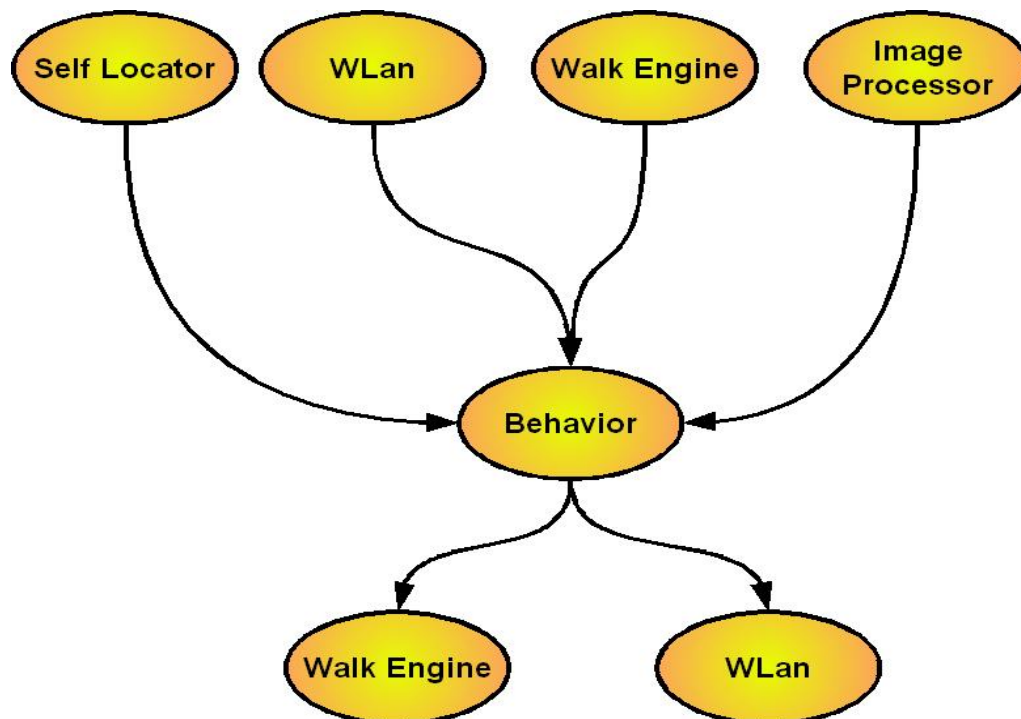
Θα πρέπει να τονιστεί σε αυτό το σημείο πως όλα τα παραπάνω δεν

συντελούν σε καμία περίπτωση αντιγραφή. Κάθε ομάδα που δεν υλοποιεί το δικό της κώδικα εξ' αρχής είναι υποχρεωμένη στην αίτηση συμμετοχής της να δηλώσει από ποια η ποιες ομάδες έχει "δανειστεί" κώδικα και σε ποιο βαθμό, καθώς και να παράσχει ένα diff αρχείο με τις διαφορές στα τμήματα κώδικα που υλοποιήθηκε από μέρους της, σε σχέση με τον αρχικό. Με τον τρόπο αυτό εξαλείφεται οποιαδήποτε υποψία αντιγραφής και νοθείας του διαγωνισμού.

Με τη λογική αυτή αποφασίστηκε να στηριχθούμε κι εμείς σαν νέα ομάδα σε κώδικα και υλοποιήσεις κάποιας άλλης ομάδας. Μετά από δοκιμές και συζητήσεις επιλέχθηκε ο κώδικας που είχε δημοσιευτεί το 2006 από την ομάδα SPQR – Legged [15] από το Πανεπιστήμιο της Ρώμης στην Ιταλία. Με τη σειρά της η ομάδα SPQR – Legged είχε βασιστεί στον κώδικα της German Team για το 2004.

5.1.2 Περιγραφή και Χαρακτηριστικά του κώδικα

Ο κώδικας της SPQR – Legged είναι δομημένος όπως φαίνεται στην Εικόνα 34, όσον αφορά το τμήμα του που είναι σχετικό με τη συμπεριφορά του ρομπότ. Κάθε βασική λειτουργία του ρομπότ (περπάτημα, εντοπισμός θέσης κ.ά.) υλοποιείται σαν ξεχωριστό τμήμα του κώδικα. Όλα τα τμήματα αλληλεπιδρούν με το κεντρικό κομμάτι του σχεδιασμού, στέλνοντάς δεδομένα κάθε χρονική στιγμή και λαμβάνοντας εντολές με την κατάλληλη ενέργεια για κάθε χρονική στιγμή. Έτσι το κομμάτι που υλοποιεί τη συμπεριφορά του ρομπότ ενημερώνεται από τον Self Locator για την εκτίμηση που έχει το ρομπότ για την ακριβή θέση του στο γήπεδο τη συγκεκριμένη χρονική στιγμή, από τον Image Processor για το τι ακριβώς βλέπει τη στιγμή εκείνη (τέρμα, μπάλα κ.ά.), από τη Walk Engine για την ακριβή στάση που έχει και από το WLaп για τυχόν μηνύματα από τους συμπαίκτες του ή τον Game Controller. Συνυπολογίζοντας όλα αυτά αποστέλλει στο δίκτυο μηνύματα για τους συμπαίκτες του, αν υπάρχουν και ειδοποιεί τη Walk Engine για το ποια κίνηση θα ακολουθήσει (σουτ, στροφή, αρπαγή της μπάλας κ.ά.). Η επικοινωνία ανάμεσα στα ξεχωριστά αυτά τμήματα κώδικα γίνεται με την ανταλλαγή ροών δεδομένων (streams).



Εικόνα 34: modules communication

5.1.3 Δομή κώδικα

Ο κώδικας που υλοποιεί το κομμάτι της συμπεριφοράς (ή σχεδιασμού) υλοποιείται σε δύο φάσεις: αρχικά υλοποιούνται οι απαραίτητες συναρτήσεις ή κλάσεις σε C++ και στη συνέχεια αυτές καλούνται μέσω των Petri Net Plans (PNP)[11]. Οι κλάσεις σε C++ δηλώνονται σε κατάλληλα αρχεία με συγκεκριμένη μορφή: κάθε κλάση περιέχει τέσσερις συναρτήσεις: `initialization()`, `executeStep()`, `termination()` και `finished()`. Ανάλογα με τον τύπο και τη λειτουργία της κλάσεις οι συναρτήσεις αυτές δέχονται και τα κατάλληλα ορίσματα. Κάθε ενέργεια του ρομπότ περιγράφεται από αυτές τις δομές.

Η συνάρτηση `initialization()` περιλαμβάνει τις όποιες αρχικοποιήσεις που απαιτούνται για να εκτελεστεί η συγκεκριμένη ενέργεια. Για παράδειγμα για να επιχειρήσει ευθύβολο σουτ το ρομπότ πρέπει να βρίσκεται σε όρθια στάση. Στην περίπτωση αυτή στη συνάρτηση `initialization()` καλείται κατάλληλη εντολή, η οποία τοποθετεί το ρομπότ στην όρθια στάση.

Η συνάρτηση `executeStep()` είναι υπεύθυνη για την εκτέλεση της συγκεκριμένης ενέργειας. Στο προηγούμενο παράδειγμα είναι υπεύθυνη για την κλήση της δομής με τη βοήθεια της οποίας σουτάρει το ρομπότ.

Η συνάρτηση `termination()` αναλαμβάνει να εκτελέσει όλες εκείνες τις ενέργειες που πιθανόν είναι απαραίτητο να εκτελεστούν μετά το πέρας της ενέργειας του ρομπότ ή αμέσως μετά τη διακοπή της ενέργειας από κάποιο `interrupt`. Στο παράδειγμα που δώσαμε η συνάρτηση `termination()` θα καλούσε την κατάλληλη συνάρτηση, ώστε το ρομπότ να επιστρέψει στην θέση ή στάση που είχε πριν εκτελέσει το σουτ.

Η συνάρτηση `finished` εξετάζει περιοδικά αν έχει τελειώσει η εκτέλεση της `executeStep()`. Αν όχι η εκτέλεση του προγράμματος συνεχίζεται από το σημείο που βρισκόταν, διαφορετικά καλείται η `termination()`.

Φυσικά είναι εμφανές ότι κάποια κλάση μπορεί να έχει κενές κάποια ή κάποιες από τις συναρτήσεις αυτές (π.χ να μην απαιτείται αρχικοποίηση) ή η συνάρτηση `finished()` να επιστρέφει πάντα την τιμή μηδέν. Αυτό μπορεί να συμβεί για παράδειγμα όταν το ρομπότ αναζητά τη μπάλα. Τότε θα πρέπει να αλλάξει κατάσταση μόνο αν δει τη μπάλα και ενεργοποιηθεί κάποιο `interrupt`, διαφορετικά δεν θα πρέπει να σταματήσει να την αναζητά.

Οι συναρτήσεις αυτές καλούνται κατ' αντιστοιχία στους κόμβους των Petri Net Plans. Έτσι όταν το token βρίσκεται στον κόμβο `start` εκτελείται η συνάρτηση `initialization()`, όταν βρίσκεται στον κόμβο `execute` εκτελείται η συνάρτηση `executeStep` και όταν βρίσκεται στον κόμβο `end` εκτελείται η συνάρτηση `termination()`.

5.1.4 Δημιουργία νέων κλάσεων – ενεργειών

Όλες οι ενέργειες που αφορούν το τμήμα της συμπεριφοράς δηλώνονται σε κατάλληλα αρχεία. Οι ενέργειες δηλώνονται με τη δομή που περιγράφηκε στην προηγούμενη παράγραφο. Στη συνέχεια τα αντικείμενα των κλάσεων που πιθανόν

απαιτείται να δηλωθούν και να χρησιμοποιηθούν δηλώνονται σε ένα αρχείο που ονομάζεται actions.cfg με τον ακόλουθο τρόπο:

```
[Class.ActionName]
```

```
# par1=val1  
# ...  
# parN=valN
```

όπου Class το όνομα της κλάσης, ActionName το όνομα του αντικειμένου της κλάσης (στην ουσία της ενέργειας που θα εκτελεστεί), parN το N – οστό όρισμα του δημιουργού (Constructor) της κλάσης και valN η τιμή που παίρνει το parN στη συγκεκριμένη περίπτωση.

Στη συνέχεια δεσμεύεται μνήμη για τα αντικείμενα που δηλώνονται με τη βοήθεια της ρουτίνας Action Loader ως εξής: στα αρχεία αυτά περιέχονται οι κλάσεις που δηλώθηκαν. Ακολουθώντας ο Action Loader διατρέχει το αρχείο actions.cfg και δεσμεύει μνήμη για κάθε αντικείμενο κλάσης που έχει δηλωθεί μέσα στον Action Loader.

5.2 Κλάσεις και αντικείμενα σε C++

Φυσικά για να καλυφθεί το εύρος της λειτουργικότητας που απαιτεί η συγκεκριμένη υλοποίηση ήταν απαραίτητη και η δημιουργία νέων ενεργειών που θα συμπλήρωναν τις ήδη υπάρχουσες των SPQR – Legged και German Team 2004. Στη συνέχεια παρατίθενται οι νέες κλάσεις – ενέργειες που υλοποιήθηκαν με μια σύντομη περιγραφή για τη λειτουργία και τη χρησιμότητά τους, καθώς και τα αντικείμενα των κλάσεων αυτών που δημιουργήθηκαν για να καλύψουν

συγκεκριμένες ενέργειες. Υπενθυμίζεται πως όλες οι κλάσεις που αντιπροσωπεύουν ενέργειες έχουν υποχρεωτικά τέσσερις συναρτήσεις (`initialization()`, `executeStep()`, `termination()` και `finished()`).

- **SendLostBall**

Όταν ένα ρομπότ χάσει τη μπάλα, στέλνει μήνυμα στο δίκτυο για να ενημερώσει και τα υπόλοιπα. Με τον τρόπο αυτό τα υπόλοιπα ρομπότ ειδοποιούνται να διαφοροποιήσουν την τακτική τους. Το μήνυμα αποστέλλεται στη συνάρτηση `initialization()`, η συνάρτηση `finished()` επιστρέφει πάντα την τιμή “ένα” (1), πράγμα που σημαίνει πως η εκτέλεση της συνάρτησης `executeStep` έχει ολοκληρωθεί και οι άλλες δύο συναρτήσεις είναι κενές.

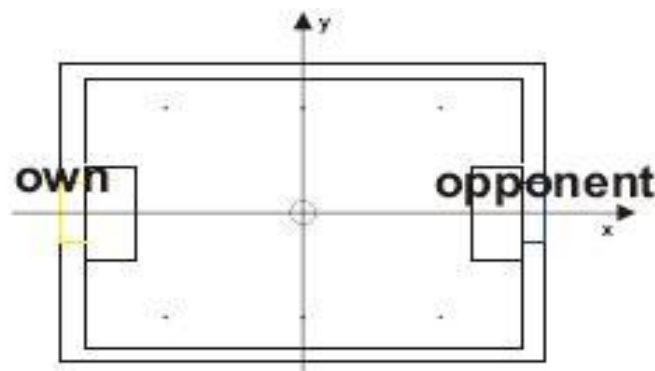
- **SendGotBall**

Όταν ένα ρομπότ κερδίσει τη μπάλα, ειδοποιεί μέσω του δικτύου και τα υπόλοιπα, ώστε να προσαρμόσουν την τακτική τους. Η λειτουργία των συναρτήσεων είναι ανάλογη της `SendLostBall`.

- **PasDefence**

Η κλάση αυτή υλοποιεί την τεχνική της άμυνας που εφαρμόζουν όλα τα ρομπότ. Με δεδομένες τις θέσεις και τον προσανατολισμό των ρομπότ μέσα στο γήπεδο σύμφωνα με το ορθοκανονικό σύστημα συντεταγμένων που φαίνεται στην εικόνα 35⁹, η κλάση αυτή τοποθετεί το ρομπότ σε ένα σημείο A (x, y) με το y να τοποθετείται ανάμεσα στη μπάλα και στο τέρμα της ομάδας του ρομπότ και x μια παράμετρος ανάλογη με το ρόλο του ρομπότ και την τακτική που ακολουθείται. Έτσι για παράδειγμα αν θέλουμε το ρομπότ να κινείται πάνω στη μεσαία γραμμή, θα δώσουμε $x = 0$. Η παράμετρος x μπορεί

να πάρει τιμές στο διάστημα $[-1500, 1500]$, δηλαδή μέσα στα όρια των τελικών γραμμών του γηπέδου. Η συνάρτηση `initialization()` δίνει κάποιες αρχικές τιμές στην ταχύτητα και τη φορά κίνησης του ρομπότ, η συνάρτηση `executeStep()` μετακινεί συνεχώς το ρομπότ ανάλογα με τη θέση της μπάλας, η συνάρτηση `finished()` επιστρέφει πάντα την τιμή “μηδέν” (0) (αφού το ρομπότ δε σταματά να αμύνεται αν δεν αλλάξει η τακτική) και η συνάρτηση `termination()` είναι κενή.



Εικόνα 35: field oriented

- **Defending**

Η λειτουργία της είναι παρόμοια με της `PasDefence`, με τη μόνη διαφορά πως η x συντεταγμένη δεν ορίζεται πια από πριν, δηλαδή το ρομπότ παίζει άμυνα σε όποιο σημείο του γηπέδου, σε σχέση με την x συντεταγμένη του, βρίσκεται εκείνη τη στιγμή. Αν λοιπόν το ρομπότ βρίσκεται στο σημείο $A(1000, 500)$, θα μετακινείται στην ευθεία γραμμή που ορίζεται από τα σημεία $A_n(1000, y)$, όπου y ένα σημείο ανάμεσα στη μπάλα και στο τέρμα της ομάδας του.

- **KouretesTrackBall**

Η κλάση αυτή υλοποιεί μια παραλλαγή της παρακολούθησης της μπάλας. Όταν η μπάλα βρίσκεται στο οπτικό πεδίο του ρομπότ τότε αυτό την παρακολουθεί στρίβοντας το κεφάλι του ή ακόμη και γυρίζοντας το σώμα του αν χρειαστεί, για χρόνο που ορίζεται από κατάλληλο όρισμα. Αν δεν δοθεί χρόνος μέσω του ορίσματος, τότε ο χρόνος ορίζεται σε δέκα δευτερόλεπτα. Η κλάση αυτή χρησιμοποιείται σε συνδυασμό με την κλάση που υλοποιεί την αναζήτηση σημαδιών στο γήπεδο για να προσδιοριστεί η θέση του ρομπότ. Ανά δέκα δευτερόλεπτα ή ανά τακτά χρονικά διαστήματα καθορισμένα από κατάλληλο όρισμα το ρομπότ σταματά να κοιτά τη μπάλα και μετακινεί το κεφάλι του στο διάστημα $[-\pi/2, \pi/2]$, ώστε να διαπιστώσει την ακριβή θέση του και αμέσως μετά επιστρέφει στη μπάλα. Η διαδικασία αυτή είναι πολύ χρήσιμη όταν το ρομπότ παρακολουθεί για μεγάλο διάστημα τη μπάλα και μέσα στο οπτικό του πεδίο δεν υπάρχει κανένα από τα χαρακτηριστικά σημάδια του γηπέδου. Η συνάρτηση `initialization()` αποθηκεύει σε μια μεταβλητή τη χρονική στιγμή που ξεκινά η εκτέλεση, η συνάρτηση `executeStep()` μετακινεί το κεφάλι και το σώμα του ρομπότ, ώστε αυτό να παρακολουθεί συνεχώς τη μπάλα, η συνάρτηση `finished()` επιστρέφει την τιμή “ένα” (1) αν έχει παρέλθει συγκεκριμένο χρονικό διάστημα και η συνάρτηση `termination()` είναι κενή.

5.3 Επικοινωνία

Όπως γίνεται φανερό από τις περιγραφές του αλγορίθμου συντονισμού που υλοποιήθηκε, καίριο ρόλο για τη σωστή λειτουργία και εφαρμογή του παίζει η ποιότητα της διεπικοινωνίας ανάμεσα στα ρομπότ. Στην ενότητα αυτή θα παρουσιαστεί η βάση κώδικα που προϋπήρχε, οι βελτιώσεις που έγιναν και τα αποτελέσματα που προέκυψαν, αφού αν και το θέμα της επικοινωνίας μοιάζει ανεξάρτητο από τη συνολική εργασία που παρουσιάζεται εδώ, είναι απαραίτητο για

την ομαλή και ολοκληρωμένη λειτουργία της υλοποίησής μας.

5.3.1 Προϋπάρχουσα δομή

Όπως έχει αναφερθεί στηριχθήκαμε στον κώδικα της ομάδας SPQR – Legged 2006 για να αναπτύξουμε τις δικές μας υλοποιήσεις. Μια από τις αρνητικές επιπτώσεις της επιλογής μας αυτής είναι και το γεγονός πως κάποιες από τις λύσεις που έχουν δοθεί μέσω του κώδικα που προϋπάρχει ίσως να μην είναι οι ενδεδειγμένες.

Έτσι λοιπόν φτάνοντας στο θέμα του δικτύου διαπιστώσαμε πως η επικοινωνία μεταξύ των ρομπότ υλοποιείται με μάλλον μη αποδοτικό τρόπο. Τα ρομπότ επικοινωνούν μεταξύ τους με το πρωτόκολλο UDP (User Datagram Protocol) [16] κάνοντας μετάδοση προς όλες τις κατευθύνσεις (Broadcast). Αποστέλλουν δηλαδή το μήνυμα σε οποιονδήποτε παραλήπτη βρίσκεται στο ίδιο υποδίκτυο (αποστέλλοντάς το στην ουσία στη διεύθυνση 255.255.255.255), κάτι που αντενδείκνυται και λόγω των κινδύνων ασφαλείας που διατρέχονται (και οι αντίπαλοι λαμβάνουν τα μηνύματα), αλλά και λόγω των περιορισμένων δυνατοτήτων της κάρτας δικτύου των ρομπότ. Μια διαφορετική υλοποίηση των πρωτοκόλλων επικοινωνίας θα ξέφευγε από τα πλαίσια και τους σκοπούς της παρούσας διπλωματικής. Για το λόγο αυτό αποφασίσαμε να βελτιώσουμε την απόδοση της υπάρχουσας υλοποίησης.

5.3.2 Δοκιμές και προβλήματα

Μετά την υλοποίηση των αλγορίθμων συντονισμού των ρομπότ κρίθηκε αναγκαία η δοκιμή της αποδοτικότητας του δικτύου. Το πείραμα που εκτελέστηκε ήταν το ακόλουθο: αν το πρώτο ρομπότ έβλεπε τη μπάλα έστελνε στο δίκτυο την τιμή “ένα” (1) και παρέμενε ακίνητο στη θέση του. Αν το δεύτερο ρομπότ λάμβανε την τιμή “ένα” (1) από το πρώτο ρομπότ, έστελνε στο δίκτυο την τιμή “δύο” (2) και παρέμενε κι αυτό ακίνητο στη θέση του. Τέλος αν το τρίτο και το τέταρτο ρομπότ λάμβαναν την

τιμή “δύο” (2) από το δεύτερο ρομπότ, αναζητούσαν τη μπάλα. Αντίστροφα αν το πρώτο ρομπότ δεν έβλεπε τη μπάλα, την αναζητούσε, και έστελνε στο δίκτυο την τιμή “δύο” (2). Αν το δεύτερο ρομπότ λάμβανε από το πρώτο την τιμή “δύο” (2), αναζητούσε τη μπάλα και έστελνε στο δίκτυο την τιμή “ένα” (1). Τέλος αν το τρίτο και τέταρτο ρομπότ λάμβαναν από το δεύτερο την τιμή “δύο” (2) παρέμεναν ακίνητα στη θέση τους.

Οι πρώτες δοκιμές με όλα τα ρομπότ ήταν αποτυχημένες. Το τρίτο και το τέταρτο ρομπότ δεν αντιδρούσαν σωστά ή ανταποκρίνονταν με πολύ μεγάλη καθυστέρηση, απαγορευτική για ένα γοργά εξελισσόμενο παιχνίδι όπως το ρομποτικό ποδόσφαιρο. Από την άλλη πλευρά όσο λιγότερα ρομπότ χρησιμοποιούνταν στο πείραμα τόσο καλύτερα ήταν τα αποτελέσματα. Το πρόβλημα εντοπίστηκε στην μικρή δυναμική των καρτών δικτύου των ρομπότ και στη διάρθρωση του κώδικα της ομάδας SPQR - Legged.

5.3.3 Προτεινόμενη λύση

Η λύση που δόθηκε στο πρόβλημα ήταν η αντιμετώπιση των ρομπότ ως κόμβους δικτύου. Κάθε ρομπότ επανεκπέμπει κάθε μήνυμα που λαμβάνει. Με τον τρόπο αυτό επιτυγχάνεται εικονική, αλλά αρκετά σημαντική ενίσχυση του σήματος. Έτσι, στο προηγούμενο πείραμα, αν το τρίτο και τέταρτο ρομπότ λάβουν κάποιο σήμα από το πρώτο ρομπότ, το οποίο δεν επηρεάζει τη λειτουργικότητα και τη συμπεριφορά τους, δεν θα το αγνοήσουν, αλλά θα το εκπέμψουν στο δίκτυο κι αυτά με τη σειρά τους. Με τον τρόπο αυτό είναι πολύ πιο πιθανό το μήνυμα να φτάσει στο δεύτερο ρομπότ, το οποίο επηρεάζει άμεσα.

Φυσικά η λύση που δόθηκε, πέρα από τις απαιτούμενες εκτεταμένες αλλαγές στο υπάρχον πρωτόκολλο που απαιτούσε, δημιουργούσε και κινδύνους deadlock, αφού ήταν δύσκολο να αναγνωριστεί το πιο πρόσφατο μήνυμα. Για παράδειγμα, αν ο επιθετικός έστελνε μήνυμα στο δίκτυο αναφέροντας ότι έχασε τη μπάλα, αλλά τα υπόλοιπα ρομπότ λάμβαναν πιο πρόσφατο μήνυμα, προερχόμενο από αναμεταδόσεις, με περιεχόμενο που ενημέρωνε ότι ο επιθετικός κέρδισε τη μπάλα, τότε οι συνέπειες στην τακτική δεν θα ήταν οι αναμενόμενες, όπως είναι λογικό. Για

το λόγο αυτό αποφασίστηκε μαζί με το περιεχόμενο του μηνύματος να αποστέλλεται και πληροφορία που να καταδεικνύει τον αρχικό αποστολέα του μηνύματος, καθώς και την τοπική ώρα του ρομπότ από το οποίο απεστάλη. Επιπλέον, κάθε ρομπότ θα κρατά τη διαφορά της ώρας του συστήματός του με την ώρα κάθε ρομπότ που του στέλνει μήνυμα.

5.3.4 Κώδικας C++

Κάθε μήνυμα περιέχει τα ακόλουθα πεδία (μετά και από τις αλλαγές που πραγματοποιήσαμε στο πρωτόκολλο):

- `timeStamp`: Η ώρα που απεστάλη το μήνυμα με βάση το ρολόι του αποστολέα.
- `lastReceivedTimeStamp`: Η ώρα που ελήφθη το τελευταίο μήνυμα από το ρομπότ με βάση το ρολόι του αποστολέα.
- `IncomingTimeStamp`: Η ώρα που ελήφθη το τελευταίο μήνυμα από το ρομπότ με βάση το δικό του ρολόι.
- `TimeOffset`: Τοπική ώρα – ώρα αποστολέα. Αποθηκεύει τη διαφορά ώρας ανάμεσα στο ρομπότ και στον αποστολέα.
- `IsActual()`: Ελέγχει αν κάποιο μήνυμα είναι πραγματικό (νέο). Αν το μήνυμα είναι παλιότερο από δύο δευτερόλεπτα αγνοείται.
- `GetTimeStampInOwnTime()`: Μετατρέπει την ώρα του λαμβανόμενου μηνύματος σε τοπική ώρα.

Επίσης κάθε μήνυμα συμπεριφοράς περιέχει τα ακόλουθα επιπλέον πεδία:

- `enum Message { lostBall, gotBall };`

Δηλώνει το περιεχόμενο του μηνύματος και αποστέλλει πληροφορία για το αν το ρομπότ έχασε ή κέρδισε τη μπάλα.

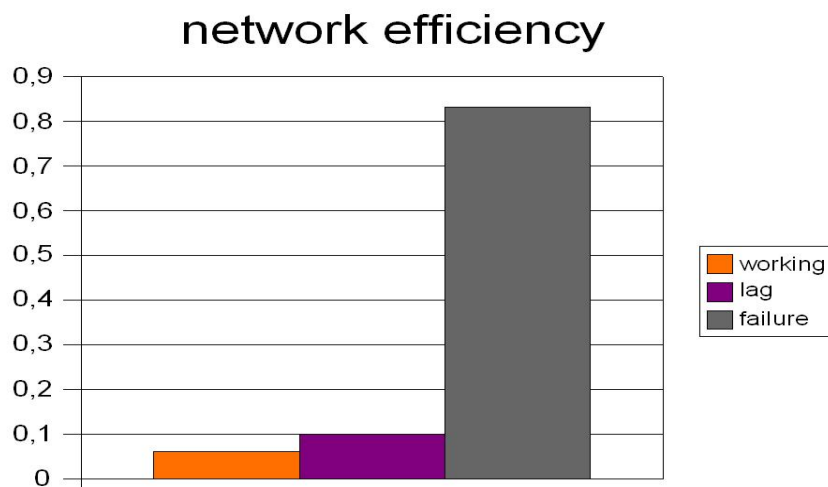
- `enum OriginalSender {keeper, defender, midfielder, attacker}`

Δείχνει ποιος είναι ο αρχικός αποστολέας του μηνύματος, πριν τις όποιες επαναεκπομπές του.

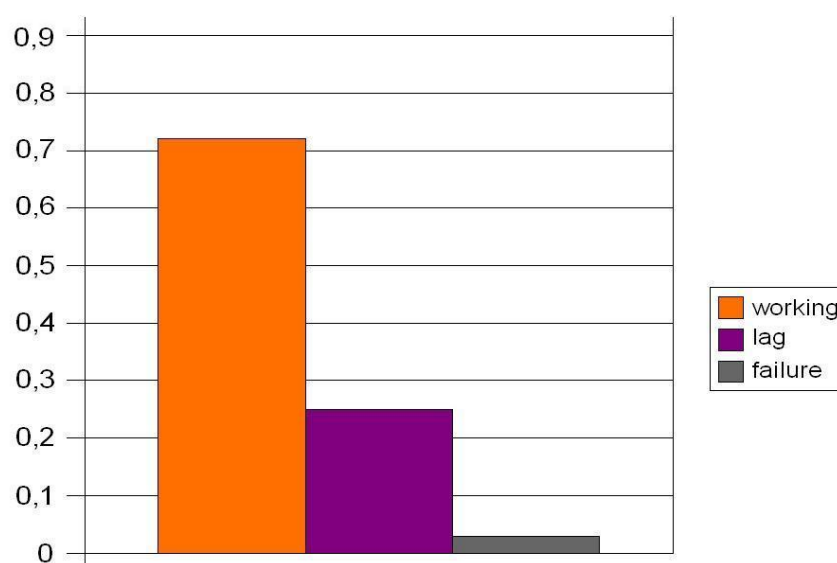
Τέλος, πρέπει να τονίσουμε πως κάθε ρομπότ αποθηκεύει τα εισερχόμενα μηνύματα σε έναν πίνακα τεσσάρων θέσεων. Επομένως για να διαπιστώσουμε ποιο μήνυμα είναι το πιο πρόσφατο εκτελούμε τις συναρτήσεις `GetTimeStampInOwnTime()` και `IsActual()` για κάθε μήνυμα του πίνακα και μετά συγκρίνουμε τα πρόσφατα `timeStamps` μεταξύ τους.

5.3.5 Αποτελέσματα

Δοκιμάζοντας τις τροποποιήσεις που πραγματοποιήσαμε στο πείραμα της παραγράφου 5.3.2 παρατηρήσαμε θεαματική βελτίωση, όπως φαίνεται και από τις εικόνες 36 και 37. Διευκρινιστικά αναφέρουμε πως ως επιτυχημένη κρίθηκε κάθε δοκιμή που είχε λογική καθυστέρηση στην ανταπόκριση των ρομπότ στα μηνύματα (δηλαδή καθυστέρηση που να μην επηρεάζει τη ροή του παιχνιδιού), λαμβάνοντας υπόψιν όμως την καθυστέρηση που παρουσιάζουν τα ρομπότ λόγω του εξοπλισμού των δοκιμών. Για παράδειγμα, αν παρατηρήσουμε μια καθυστέρηση της τάξης των δύο δευτερολέπτων από τη στιγμή που ο `Game Controller` δώσει μια εντολή, μέχρι να ανταποκριθεί το ρομπότ, τότε θεωρούμε την καθυστέρηση αυτή δεδομένη για το συγκεκριμένο σύστημα. Έτσι αν στο πείραμά μας κάποιο ρομπότ στείλει μήνυμα σε ένα άλλο και το δεύτερο ανταποκριθεί μετά από δύο δευτερόλεπτα, θεωρούμε την καθυστέρηση αυτή δεδομένη και οφειλόμενη στην δεδομένη καθυστέρηση του δικτύου.



Εικόνα 36: Απόδοση του δικτύου πριν τις βελτιώσεις



Εικόνα 37: Απόδοση του δικτύου μετά τις βελτιώσεις

Στα παραπάνω διαγράμματα η τρίτη στήλη απεικονίζει το ποσοστό των δοκιμών, επί των συνολικών δοκιμών που πραγματοποιήθηκαν, που ήταν αποτυχημένες, δηλαδή κάποιο από τα ρομπότ δεν έλαβε τα μηνύματα που έπρεπε.

Η δεύτερη στήλη απεικονίζει το ποσοστό των δοκιμών, στις οποίες όλα τα ρομπότ έλαβαν τα μηνύματα που έπρεπε να λάβουν από τον ορισμό του πειράματος, αλλά με πολύ μεγάλη καθυστέρηση. Τέλος, η τρίτη στήλη απεικονίζει το ποσοστό των δοκιμών, στις οποίες το σύστημα λειτούργησε με επιτυχία.

,

6 Συμπεράσματα

Για να λειτουργήσει τέλεια το μοντέλο που προτείνουμε πρέπει να υποστηρίζεται από πολύ καλό εντοπισμό θέσης από μέρους των ρομπότ, αλλά και από πολύ καλή λειτουργία της όρασης των ρομπότ. Σε όλες τις δοκιμές που πραγματοποιήσαμε παρ' όλα αυτά, ήταν εμφανής η συνεργασία από μέρους των ρομπότ και ο πιο ανεπτυγμένος και πιο πλήρης τρόπος παιχνιδιού σε σχέση με τις υλοποιήσεις άλλων ομάδων. Φυσικά επειδή όλα αυτά είναι δύσκολο να φανούν σε μια αναφορά, μέρος των συμπεριφορών που υλοποιήθηκαν υπάρχουν στην ηλεκτρονική διεύθυνση www.intelligence.tuc.gr/kouretes.

Η παρούσα εργασία είχε σαν σκοπό να προτείνει ένα πιο ουσιαστικό μοντέλο πολυπρακτορικής συνεργασίας στην κατηγορία τετραπόδων του RoboCup. Φυσικά οι αλγόριθμοι και οι πρακτικές που εφαρμόζονται μπορούν να επεκταθούν και σε άλλους τομείς, όπως η συνεργατική πλανητική εξερεύνηση και η έρευνα και διάσωση. Μάλιστα επειδή θεωρήθηκε πως δεν έχει πραγματοποιηθεί κάτι ανάλογο στο χώρο, η διπλωματική αυτή εργασία οδήγησε και σε μια δημοσίευση στο 19th International Conference on Tools with Artificial Intelligence (ICTAI). Επιπλέον προτείναμε έναν πολύ αποδοτικό και εύκολο τρόπο βελτίωσης της ασύρματης επικοινωνίας σε πολυπρακτορικά συστήματα.

Τέλος τρία πολύ γενικά συμπεράσματα που εξάγονται από την τριβή μας με το RoboCup, αλλά και από την εκπόνηση της παρούσας διπλωματικής εργασίας είναι ότι η συμπεριφορά των ρομπότ μπορεί να είναι εξαιρετικά απρόβλεπτη, είναι προτιμότερο και πιο επωφελές μακροπρόθεσμα να μην στηρίζεις σε κώδικα και υλοποιήσεις άλλων και ο νόμος του Murphy ισχύει πάντα.

Στο μέλλον κάποιες αλλαγές που θα οδηγούσαν σε καλύτερα αποτελέσματα θα ήταν η αντικατάσταση του UDP πρωτοκόλλου επικοινωνίας από TCP [17] και η τελειοποίηση όλων των στοιχειωδών ενεργειών (όπως περπάτημα, αρπαγή της μπάλας κ.ά.) με ενισχυτική μάθηση. Η ενισχυτική μάθηση επίσης, θα μπορούσε να χρησιμοποιηθεί και για τη συνολική βελτίωση και συνεργασία της ομάδας είτε εφαρμόζοντας συνεργατική πολυπρακτορική μάθηση [18] (όπου πολλοί πράκτορες μαθαίνουν να συνεργάζονται σαν ομάδα) είτε με τη μορφή της ανταγωνιστικής πολυπρακτορικής μάθησης [19] (όπου δύο ομάδες πρακτόρων μαθαίνουν να

ανταγωνίζονται η μία την άλλη, αλλά τα μέλη της κάθε ομάδας να συνεργάζονται μεταξύ τους.

Βιβλιογραφία

- 1: H. Kitano, M. Asada, Y. Kuniyoshi, I. Noda, E. Osawa, , and H. Matsubara. Robocup: A challenge problem for AI. *AI Magazine*, 18(1):73–85, 1997
- 2: Four - Legged Rules 2007, RoboCup Federation 2007, www.robocup.org
- 3: Teams, <http://www.tzi.de/4legged/bin/view/Website/Teams2007>,
- 4: Kazutaka Takeshita, Takashi Okuzumi, Syota Kase, Yuji Hasegawa, Hisanori Mitsumoto, Ryuichi Ueda, Kazunori Umeda, Hisashi Osumi, and Tamio Arai, Technical Report of Team ARAIBO, 2007
- 5: Röfer, T., Brunn, R., Czarnetzki, S., Dassler, M., Hebbel, M., Jüngel, M., Kerkhof, T., Nistico, W., Oberlies, T., Rohde, C., Spranger, M., Zarges, C. , GermanTeam 2005, 2005
- 6: Röfer, T., Brose, J., Carls, E., Carstens, J., Göhring, D., Jüngel, M., Laue, T., Oberlies, T., Oesau, S., Risler, M., Spranger, M., Werner, C., Zimmer, J. , GermanTeam 2006, 2006
- 7: H. Levent Akın. Çetin Mericli. Baris Gokce. Fuat Geleri, Nuri Tasdemir, Buluc Celik, Melih Celik, Cerberus'06 Team Report,
- 8: Michael J. Quinlan, Naomi Henderson, Richard H. Middleton, Steven P. Nicklin, Robin Fisher, Florian Knorn, Stephan K. Chalup, Robert King, The 2006 NUbots Team Report,
- 9: Henry Work, Prof. Eric Chown, Jesse Butterfield, Ferd Convery, Patrick Costello, Jeremy Fishman, Tracy McKay, Brendan Mortimer, Matthew Murchison, Quentin Reeve, George Slavov, Johannes Strom, Yi Zhuang, Northern Bites 2006 Team Report,
- 10: Formation, [http://en.wikipedia.org/wiki/Formation_\(football\)](http://en.wikipedia.org/wiki/Formation_(football))
- 11: V. A. Ziparo and L. Iocchi. Petri net plans. In *ATPN/ACSD Fourth International Workshop on Modelling of Objects, Components, and Agents*, Finland, 2006.
- 12: T. Murata. Petri nets: Properties, analysis and applications. *Proceedings of the IEEE*, 77(4):541–580, April 1989.
- 13: J. L. Peterson. *Petri Net Theory and the Modeling of Systems.*, Prentice Hall, Upper Saddle River, NJ, 1981.
- 14: Jarp Petri Nets Analyzer, jarp.sourceforge.net
- 15: SPQR -Legged, http://www.dis.uniroma1.it/~spqr/cms/index.php?option=com_frontpage&Itemid=1
- 16: User Datagram Protocol, http://en.wikipedia.org/wiki/User_Datagram_Protocol
- 17: TCP Protocol, <http://el.wikipedia.org/wiki/TCP/IP>
- 18: C. Guestrin, M. G. Lagoudakis, and R. Parr. Coordinated reinforcement learning. In *Proceedings of the 19th International Conference on Machine Learning (ICML-2002)*, pages 227–234, 2002.
- 19: M. G. Lagoudakis and R. Parr. Learning in team markov games using factored value functions. In *Proceedings of NIPS*2002: Neural Information Processing Systems: Natural and Synthetic*, pages 1659–1666, 2002.
- 20: www.robocup.org

