

مجلة جامعة البعث

سلسلة العلوم الهندسية الميكانيكية و
الكهربائية و المعلوماتية



مجلة علمية محكمة دورية

المجلد 44 . العدد 12

1443 هـ . 2022 م

الأستاذ الدكتور عبد الباسط الخطيب

رئيس جامعة البعث

المدير المسؤول عن المجلة

رئيس هيئة التحرير

أ. د. ناصر سعد الدين

رئيس التحرير

أ. د. هائل الطالب

مديرة مكتب مجلة جامعة البعث

بشرى مصطفى

عضو هيئة التحرير	د. محمد هلال
عضو هيئة التحرير	د. فهد شريباتي
عضو هيئة التحرير	د. معن سلامة
عضو هيئة التحرير	د. جمال العلي
عضو هيئة التحرير	د. عباد كاسوحة
عضو هيئة التحرير	د. محمود عامر
عضو هيئة التحرير	د. أحمد الحسن
عضو هيئة التحرير	د. سونيا عطية
عضو هيئة التحرير	د. ريم ديب
عضو هيئة التحرير	د. حسن مشرقي
عضو هيئة التحرير	د. هيثم حسن
عضو هيئة التحرير	د. نزار عبشي

تهدف المجلة إلى نشر البحوث العلمية الأصيلة، ويمكن للراغبين في طلبها

الاتصال بالعنوان التالي:

رئيس تحرير مجلة جامعة البعث

سورية . حمص . جامعة البعث . الإدارة المركزية . ص . ب (77)

. هاتف / فاكس : 2138071 31 963 ++

. موقع الإنترنت : www.albaath-univ.edu.sy

. البريد الإلكتروني : [magazine@ albaath-univ.edu.sy](mailto:magazine@albaath-univ.edu.sy)

ISSN: 1022-467X

شروط النشر في مجلة جامعة البعث

الأوراق المطلوبة:

- 2 نسخة ورقية من البحث بدون اسم الباحث / الكلية / الجامعة + CD / word من البحث منسق حسب شروط المجلة.
 - طابع بحث علمي + طابع نقابة معلمين.
 - إذا كان الباحث طالب دراسات عليا:
يجب إرفاق قرار تسجيل الدكتوراه / ماجستير + كتاب من الدكتور المشرف بموافقة على النشر في المجلة.
 - إذا كان الباحث عضو هيئة تدريسية:
يجب إرفاق قرار المجلس المختص بإنجاز البحث أو قرار قسم بالموافقة على اعتماده حسب الحال.
 - إذا كان الباحث عضو هيئة تدريسية من خارج جامعة البعث :
يجب إحضار كتاب من عمادة كليته تثبت أنه عضو بالهيئة التدريسية و على رأس عمله حتى تاريخه.
 - إذا كان الباحث عضواً في الهيئة الفنية :
يجب إرفاق كتاب يحدد فيه مكان و زمان إجراء البحث , وما يثبت صفته وأنه على رأس عمله.
 - يتم ترتيب البحث على النحو الآتي بالنسبة لكليات (العلوم الطبية والهندسية والأساسية والتطبيقية):
عنوان البحث .. ملخص عربي و إنكليزي (كلمات مفتاحية في نهاية الملخصين).
- 1- مقدمة
 - 2- هدف البحث
 - 3- مواد وطرق البحث
 - 4- النتائج ومناقشتها .
 - 5- الاستنتاجات والتوصيات .
 - 6- المراجع.

- يتم ترتيب البحث على النحو الآتي بالنسبة لكليات (الآداب - الاقتصاد - التربية - الحقوق - السياحة - التربية الموسيقية وجميع العلوم الإنسانية):
- عنوان البحث .. ملخص عربي و إنكليزي (كلمات مفتاحية في نهاية الملخصين).

1. مقدمة.
2. مشكلة البحث وأهميته والجديد فيه.
3. أهداف البحث و أسئلته.
4. فرضيات البحث و حدوده.
5. مصطلحات البحث و تعريفاته الإجرائية.
6. الإطار النظري و الدراسات السابقة.
7. منهج البحث و إجراءاته.
8. عرض البحث و المناقشة والتحليل
9. نتائج البحث.
10. مقترحات البحث إن وجدت.
11. قائمة المصادر والمراجع.

7- يجب اعتماد الإعدادات الآتية أثناء طباعة البحث على الكمبيوتر:

- أ- قياس الورق 25×17.5 B5.
 - ب- هوامش الصفحة: أعلى 2.54- أسفل 2.54 - يمين 2.5- يسار 2.5 سم
 - ت- رأس الصفحة 1.6 / تذييل الصفحة 1.8
 - ث- نوع الخط وقياسه: العنوان . Monotype Koufi قياس 20
- . كتابة النص Simplified Arabic قياس 13 عادي . العناوين الفرعية Simplified Arabic قياس 13 عريض.

- ج . يجب مراعاة أن يكون قياس الصور والجدول المدرجة في البحث لا يتعدى 12سم.
- 8- في حال عدم إجراء البحث وفقاً لما ورد أعلاه من إشارات فإن البحث سيهمل ولا يرد البحث إلى صاحبه.
- 9- تقديم أي بحث للنشر في المجلة يدل ضمناً على عدم نشره في أي مكان آخر , وفي حال قبول البحث للنشر في مجلة جامعة البعث يجب عدم نشره في أي مجلة أخرى.
- 10- الناشر غير مسؤول عن محتوى ما ينشر من مادة الموضوعات التي تنشر في المجلة

11- تكتب المراجع ضمن النص على الشكل التالي: [1] ثم رقم الصفحة ويفضل استخدام التهميش الإلكتروني المعمول به في نظام وورد WORD حيث يشير الرقم إلى رقم المرجع الوارد في قائمة المراجع.

تكتب جميع المراجع باللغة الانكليزية (الأحرف الرومانية) وفق التالي:
آ . إذا كان المرجع أجنبياً:

الكنية بالأحرف الكبيرة . الحرف الأول من الاسم تتبعه فاصلة . سنة النشر . وتتبعها معترضة (-) عنوان الكتاب ويوضع تحته خط وتتبعه نقطة . دار النشر وتتبعها فاصلة . الطبعة (ثانية . ثالثة) . بلد النشر وتتبعها فاصلة . عدد صفحات الكتاب وتتبعها نقطة .
وفيما يلي مثال على ذلك:

-MAVRODEANUS, R1986- Flame Spectroscopy. Willy, New York, 373p.

ب . إذا كان المرجع بحثاً منشوراً في مجلة باللغة الأجنبية:

. بعد الكنية والاسم وسنة النشر يضاف عنوان البحث وتتبعه فاصلة, اسم المجلد ويوضع تحته خط وتتبعه فاصلة . المجلد والعدد (كتابية مختزلة) وبعدها فاصلة . أرقام الصفحات الخاصة بالبحث ضمن المجلة.
مثال على ذلك:

BUSSE,E 1980 Organic Brain Diseases Clinical Psychiatry News ,
Vol. 4. 20 – 60

ج . إذا كان المرجع أو البحث منشوراً باللغة العربية فيجب تحويله إلى اللغة الإنكليزية و
التقيد

بالبنود (أ و ب) ويكتب في نهاية المراجع العربية: (المراجع In Arabic)

رسوم النشر في مجلة جامعة البعث

1. دفع رسم نشر (20000) ل.س عشرون ألف ليرة سورية عن كل بحث لكل باحث يريد نشره في مجلة جامعة البعث.
2. دفع رسم نشر (50000) ل.س خمسون الف ليرة سورية عن كل بحث للباحثين من الجامعة الخاصة والافتراضية .
3. دفع رسم نشر (200) مئتا دولار أمريكي فقط للباحثين من خارج القطر العربي السوري .
4. دفع مبلغ (3000) ل.س ثلاثة آلاف ليرة سورية رسم موافقة على النشر من كافة الباحثين.

المحتوى

الصفحة	اسم الباحث	اسم البحث
54-11	م. طارق عيسى أكرم المرعي ناصر أبو صالح	تطوير خوارزمية لمزامنة الملفات الموزعة في التخزين السحابي عند حدوث تعديل في المحتوى بين طرفين
84-45	رضوان المحمد د. مسعود الأتاسي	تطوير خوارزمية لمزامنة الملفات الموزعة في التخزين السحابي عند حدوث تعديل في المحتوى بين طرفين
122-85	م. رغده الإبراهيم د. شفيق باصيل د. بسام عطيه	دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات المقادة باستخدام التحكم الشعاعي
150-123	د. محمد علي محمد روان نصر	إطار عمل لتخمين جهود فريق العمل في تطوير مشاريع الويب

تطوير خوارزمية لمزامنة الملفات الموزعة في التخزين السحابي عند حدوث تعديل في المحتوى

بين طرفين

م. طارق عيسى

كلية الهندسة المعلوماتية - جامعة البعث

إشراف الدكتور: أكرم المرعي + ناصر أبو صالح

الملخص:

غالباً يتم تحقيق التخزين السحابي عالي الإتاحة باستخدام أنظمة موزعة معقدة ومتعددة المستويات مبنية على مجموعات من المخدمات ومحركات الأقراص، هناك حاجة إلى تقنيات متطورة للإدارة وموازنة الحمل والاسترداد لتحقيق أداء وإتاحة عاليين وسط وفرة من مصادر الأعطال التي تشمل البرامج والأجهزة والاتصال بالشبكة ومشكلات الطاقة، ولكن تبقى مشكلة تزامن الملفات بين المخدمات أو بين الزبون والمخدم من أعقد المشاكل حيث تعتمد الحلول التقليدية المتبعة حالياً في مختلف الأنظمة الموزعة وأنظمة التخزين السحابي كما في Google drive، Dropbox على عملية رفع أو إرسال كامل الملف عند حدوث تعديل في محتوى الملف سواء كان هذا التعديل كبير أو صغير جداً وبالتالي استهلاك كبير في موارد الشبكة وضغط على عرض الحزمة المتوفر وخصوصاً في حالات الدفع مقابل حجم الإرسال، لذلك كان لابد من البحث عن آلية أو خوارزمية تعمل على التقليل من حجم البيانات المرسلة على الشبكة.

فقد قمنا في هذا البحث بتطوير آلية جديدة لمزامنة الملفات من خلال تقسيم الملف إلى بلوكات والبحث عن مواضع التعديل وتفاذي إرسال كامل الملف، والاستعاضة عنها بإرسال أجزاء فقط من الملف وعملنا على تحقيق الخوارزمية باستخدام لغة الجافا JAVA وتقييم عملها بناء على مجموعة من المعايير.

الكلمات المفتاحية:

التخزين السحابي، الأنظمة الموزعة، Google drive، Dropbox، نموذج العمل.

Development of an algorithm to synchronize files distributed in cloud storage when content modification occurs between two sides

Abstract:

High availability cloud storage is often achieved using complex, multi-tiered distributed systems built on clusters of servers and drives. Advanced management, load balancing, and recovery techniques are needed to achieve high performance and availability amid an abundance of fault sources including software, hardware, network connectivity, and power issues, but the The problem of file synchronization between servers or between the client and the server is one of the most complex problems. The traditional solutions currently used in various distributed and cloud storage systems, as in Google drive, Dropbox, depend on the process of uploading or sending the entire file when a modification occurs in the file's content, whether this modification is large or It is very small and thus a large consumption of network resources and

pressure on the available packet width, especially in cases of payment for the size of the transmission, so it was necessary to search for a mechanism or algorithm that works to reduce the volume of data sent on the network.

In this research, we have developed a new mechanism for synchronizing files by dividing the file into blocks, searching for modification locations and avoiding sending the entire file, and replacing it with sending only parts of the file. We worked on achieving the algorithm using JAVA and evaluating its work based on a set of criteria.

Key words:

Cloud storage, Distributed system, Google drive, Dropbox, work template

1 - مقدمة:

يعرف التخزين السحابي الموزع بأنه بنية تحتية مشتتة جغرافياً، تتم من خلال نشر بيانات الأشخاص عبر الشبكة، حيث يسمح بوضع البيانات بالقرب من المستخدم النهائي، وبالتالي يسرع عمليات النقل ويقلل من ازدحام الشبكة وخطر فقدان البيانات. بالإضافة إلى ذلك وعلى عكس نماذج السحابة المركزية، لا تعتمد السحابة الموزعة على مراكز البيانات المركزية.

تعتبر السحابة الموزعة تطبيقاً لتقنيات الحوسبة السحابية لربط البيانات والتطبيقات المقدمة من مواقع جغرافية متعددة، يعني التوزيع في سياق تكنولوجيا المعلومات (IT) أن شيئاً ما يتم مشاركته بين أنظمة متعددة قد تكون أيضاً في مواقع مختلفة. تعمل السحابة الموزعة على تسريع الاتصالات للخدمات العالمية وتمكين اتصالات أكثر استجابة لمناطق معينة.

يستخدم مزودو السحابة النموذج الموزع لتمكين زمن انتقال أقل وتوفير أداء أفضل للخدمات السحابية. خارج سياق مزود السحابة، هناك مثالان آخران على السحابة الموزعة هما حوسبة الموارد العامة `public resource computing` وسحابة المتطوعين `volunteer cloud`.

السحابة الموزعة تعني سحابة أكثر خصوصية (كل ملف مقسم، مشفر من طرف إلى طرف ثم ينتشر عبر الشبكة)، أكثر أماناً (بالاعتماد على أنظمة متعددة فهو أقل عرضة للخطر) وأكثر مراعاة للبيئة.

وتعد مشكلة التزامن من أهم القضايا المدروسة حالياً ففي التخزين السحابي الموزع وهناك حاجة لإيجاد آليات قادرة على مزامنة الملف (الملفات) بين المخدمات أو بين المخدم والزيون، حيث الهدف الرئيسي الأهم هو الحصول على تبادل بيانات أقل في عملية التزامن. لكن نظم التخزين السحابي التقليدية تعمل على تبادل الملفات وإعادة ارسالها بشكل كامل عند كل عملية تعديل.

حيث يتضمن بحثنا أولاً تحديد هدف البحث ومواد وطرائق البحث ومجموعة من المفاهيم النظرية يليها النتائج والمناقشة متضمنة توصيف الخوارزمية المقترحة والمعايير المستخدمة في تقييم عملها وأدائها وفي النهاية مجموعة من النتائج والتوصيات التي حصلنا عليها.

2 - هدف البحث:

يهدف هذا البحث الى تطوير خوارزمية قادرة على مزامنة الملفات بين طرفين والحصول على نسخة متطابقة من الملف (الملفات) بينهما، وذلك تقادياً لعملية إرسال كامل الملف عند كل عملية تعديل في محتوى الملف، بالاستفادة من تقسيم الملف الى بلوكات أصغر والتعامل مع الملف كأنه مجموعة من الملفات الأصغر وتحديد موضع التعديل الحاصل وبالتالي القيام بإرسال فقط الجزء (الأجزاء) الذي تم تعديلها. وبذلك نقلل من حجم استهلاك عرض الحزمة بين الطرفين مع عدد معين من العمليات الحسابية عند الطرف المالك للنسخة الاقدم.

3 - مواد وطرائق البحث:

قمنا في هذا البحث بالاستفادة من لغة الجافا في نمذجة عمل الخوارزمية المقترحة وتطبيق عملها على عدد من الملفات باستخدام برنامج netbeans 12.5 ونسخة الجافا jdk-11.0.13 والقيام بعدد من التجارب ورسم النتائج من خلال برنامج Gnuplot version 5.0المسؤول عن رسم المخططات البيانية.

4 - المفاهيم النظرية:

4.1 - الحوسبة السحابية مقابل السحابة الموزعة:

في الحقيقة يمكن أن نقول عن الحوسبة السحابية والسحابة الموزعة أنهما نفس المفهوم لكنهم يستخدمون أنظمة مختلفة لتحقيق ذلك، تتطلب الحوسبة السحابية مركز بيانات به العديد من المخدمات للعمل عبر مهام متعددة للمستخدمين، مثل تخزين البيانات ومعالجتها وإدارتها، بينما توزع الحوسبة الموزعة المهام عبر شبكتها على أجهزة حاسوب فردية.

عند المقارنة بين المفهومين، هناك فائدة رئيسية واحدة للسحابة الموزعة: الموثوقية، نظراً لأن السحابة الموزعة تعتمد على العديد من الأنظمة بدلاً من نظام واحد ففي حالة حدوث خلل فني تكون بياناتك أكثر أماناً علاوة على ذلك فهي أسرع حيث يتم تقسيم كل مهمة وتنفيذها بواسطة العديد من أجهزة الحاسوب في وقت واحد.

4.2 - الحوسبة السحابية الموزعة مقابل الحوسبة الطرفية:

الحوسبة الطرفية مثال على السحابة الموزعة حيث تعني الحوسبة الطرفية حرفياً الحوسبة التي تحدث على حافة الشبكة، والتي تحدث على مصادر البيانات تماماً أو بالقرب منها.

تعمل الحوسبة الطرفية على تفرغ البيانات إلى السحابة أثناء فترات الذروة في حركة مرور الحوسبة لضمان السرعة والموثوقية أما السحابة الموزعة بدلاً من ذلك مهمتها الحساب والتخزين و تحقيق الاتصالات في سحابة صغيرة موجودة عبر الشبكة.

4.3 - التخزين السحابي:

يعتبر التخزين السحابي نموذج من نماذج تخزين البيانات، حيث يكون هذا التخزين تخزين فيزيائي على السحابة (مجموعة من المخدمات في موقع أو عدة مواقع بعيدة يتم إدارتها من قبل شركة مستضيفة)، مهمة التخزين السحابي بالنسبة للبيانات هو جعلها متوفرة مع إمكانية وصول إليها، أما بالنسبة للبيئة الفيزيائية فمهمتها أن تكون محمية وقيد التشغيل دائماً [4] .

مفهوم التخزين السحابي يعني أن المستخدم سيقوم بتخزين البيانات الخاصة به على السحابة عوضاً عن تخزينها على النظام المحلي، ويتم الوصول إلى هذه البيانات عن طريق اتصال شبكي بين المستخدم و المخدم (client - server).

تتميز السحابة بالخصائص الأساسية الأربع التالية:

1- المرونة: يتم توفير الخدمة أو التخزين عند الطلب عندما يحتاج المستخدم إلى مزيد من الموارد، وسيتم توفيرها تلقائياً، من ناحية أخرى سيتم تخفيض الخدمة إذا لم يكن المستخدم بحاجة إليها.

2- توفير الخدمة الذاتية وإلغاء التوفير التلقائي: يمكن للمستخدمين طلب أي مساحة من التخزين مباشرةً.

3- واجهات برمجة التطبيقات (APIs): تعتبر الواجهات القياسية ميزة لأن التطبيقات ومصادر البيانات يمكن أن تتصل مع بعضها البعض بسهولة.

4- الفواتير بناءً على استخدام الخدمة: تدفع الشركات بقدر ما تستخدم.

4.4 - البنية التحتية للتخزين الفيزيائي:

بداية كان التخزين مرتبط بالمخدمات وأجهزة الحاسوبية العملاقة والمركزية، وتستخدم التطبيقات التي تعمل على نفس الأجهزة هذا التخزين، بعد ذلك ظهرت شبكة منطقة التخزين (Storage Area Network)، وأصبح التخزين منفصلاً ولكنه متصل باستخدام اتصالات شبكية عالية الأداء، ثم تم التركيز على كلاً من البيانات والبنية التحتية الفيزيائية من خلال تقسيم هذه التخزين، مما أدى إلى فوائد عديدة مثل تقليل التكلفة والمرونة.

4.5 - التخزين كخدمة:

يمكن أن يُمنح التخزين على السحابة كخدمة للمستخدمين بدلاً من الاضطرار إلى شراء سعة تخزين فعلية بشكل محلي، زادت شعبية هذه الخدمة لأنها توفر حلاً رخيصاً للنسخ الاحتياطي والتكرار واستعادة البيانات بعد حدوث الكوارث، وبالتالي يتم تأجير التخزين باستخدام سعر عند الطلب من قبل مزودي السحابة مما يقلل من التكاليف التشغيلية.

هناك مجموعة كبيرة ومتنوعة من مزودين التخزين السحابي الذين يوفر برنامج مستخدم لاستخدام التخزين عبر الشبكة. يسمح بعضها للمستخدمين بتخزين جميع

أنواع البيانات. في حين أن البعض الآخر مخصص فقط لرسائل البريد الإلكتروني أو الصور الرقمية حيث يتحمل المزودين مسؤولية توفير خطة التوفير وصيانة مئات أو آلاف من مخدّمات البيانات للتأكد من أن جميع بيانات المستخدمين ستكون متاحة في أي وقت.

يوجد العديد من الفوائد لتخزين البيانات في السحابة على التخزين المحلي [6]:

1- تدفع الشركات فقط مقابل التخزين الذي تستخدمه، أي أن النفقات هي نفقات التشغيل فقط.

2- يمكن الوصول إلى البيانات بسرعة ويمكن الاعتماد عليها توجد البيانات على الويب عبر أنظمة تخزين متعددة بدلاً من موقع محلي.

3- حماية أفضل في حالة وقوع كارثة ما، ففي بعض الأحيان يكون لدى الشركة نسخة احتياطية محلية وفي حالات الحريق أو الكوارث الطبيعية لن يكون النسخ الاحتياطي متاحاً.

4- يوفر مزودو السحابة وفرة في العتاد وتجاوز فشل التخزين التلقائي، يساعد هذا في تجنب انقطاع الخدمة بسبب فشل الأجهزة. يعرف المزودون كيفية توزيع النسخ للتخفيف من أي فشل في الأجهزة.

5- ساعات تخزين غير محدودة عملياً، إذا لم يضطر المستخدم إلى استخدام التخزين الإضافي فسوف تتخفض التكاليف.

6- توازن عبء العمل، يساعد مزودو السحابة المستخدمين على تحقيق أفضل أداء من خلال موازنة أعباء العمل.

7- عرض موحد للتخزين، يوفر مزودو السحابة تصديراً للحصول على عرض موحد لاستخدام التخزين.

من ناحية أخرى، هناك العديد من المساوئ لتخزين البيانات باستخدام السحابة على التخزين المحلي:

1- قلة الخبرة: كان على المزودين إعادة كتابة الحلول لحل بعض حالات عدم التوافق مع تخزين البيانات عبر الإنترنت، وقد شكّل ذلك صعوبة للشركات.

2- السعر والموثوقية: يجب على المستخدم حساب فعالية التكلفة للسحابة مقابل استضافة بياناته وصيانتها.

3- الأمان: هناك احتمال أن تتم سرقة البيانات أو عرضها من قبل أشخاص غير مصرح لهم.

- 4- حدود النطاق الترددي (عرض الحزمة): إذا لم يكن النطاق الترددي بالسرعة التي يحتاجها المستخدم، فلن يكون الحل مناسباً. حيث أن النطاق الترددي هو مقياس لمقدار البيانات التي يمكن نقلها من مكان إلى آخر في فترة زمنية معينة.
- 5- مسافة الشبكة (التأخير): يؤثر مجموع التأخيرات الزمنية في الانتشار ونقل الحزم داخل الشبكة على عملية التخزين السحابي.

4.6 - النموذج المرجعي للتخزين السحابي:

من المهم أن تدعم أي واجهة للتخزين السحابي مجموعة من السمات والتي سبق وأن تحدثنا عنها: الدفع حسب الاستخدام، المرونة وبساطة الاستخدام والإدارة، مع السماح بالعديد من حالات الأعمال والعروض لفترة طويلة في المستقبل.

يُظهر النموذج الذي تم إنشاؤه ونشره بواسطة Storage Networking Industry Association (SNIA) أنواعاً متعددة من واجهات تخزين البيانات السحابية قادرة على دعم التطبيقات القديمة والجديدة، تسمح جميع الواجهات بتوفير التخزين عند الطلب والمستمدة من مجموعة من الموارد، يتم سحب السعة من مجموعة من السعات التخزينية التي توفرها خدمات التخزين. يتم تطبيق خدمات

البيانات على عناصر البيانات الفردية على النحو الذي يحدده الوصف للنظام. يحدد هذا الوصف متطلبات البيانات على أساس عناصر البيانات الفردية أو مجموعات البيانات (الحاويات containers).

واجهة إدارة البيانات السحابية (Cloud Data Management Interface) هي الواجهة الوظيفية التي ستستخدمها التطبيقات لإنشاء عناصر البيانات واستردادها وتحديثها وحذفها من السحابة، كجزء من هذه الواجهة سيتمكن المستخدم من اكتشاف إمكانات عرض التخزين السحابي واستخدام هذه الواجهة لإدارة المجموعات والبيانات الموضوعية فيها، بالإضافة إلى ذلك يمكن تعيين البيانات الوصفية على الحاويات وعناصر البيانات الموجودة بها من خلال هذه الواجهة. من المتوقع أن تكون الواجهة قادرة على التنفيذ بواسطة غالبية عروض التخزين السحابي الحالية اليوم، يمكن القيام بذلك باستخدام ملائم adapter لواجهة الملكية الموجودة أو عن طريق تنفيذ الواجهة مباشرة، بالإضافة إلى ذلك يمكن تكييف مكتبات المستخدمين الحالية مثل طريقة الوصول القابلة للتمديد (XAM Extendable Access Methods) مع هذه الواجهة.

تُستخدم CDMI بواسطة التطبيقات الإدارية والتنظيمية لإدارة الحاويات والحسابات والوصول إلى الأمان ومعلومات المراقبة / الفواتير وحتى للتخزين الذي يمكن الوصول إليه بواسطة البروتوكولات الأخرى. يتم الكشف عن إمكانات خدمات التخزين والبيانات الأساسية حتى يتمكن المستخدمين من فهم العرض، قد تقدم عروض السحابة المتوافقة مجموعة فرعية من أي واجهة طالما أنها تكشف القيود في جزء القدرات من الواجهة.

4.7 - واجهة برمجة تطبيقات التخزين السحابي (API):

تعد واجهة برمجة تطبيقات التخزين السحابي (API) طريقة للوصول إلى نظام التخزين السحابي واستخدامه، أكثر هذه الأنواع شيوعاً هي REST (نقل الحالة التمثيلية REpresentational State Transfer) على الرغم من وجود أنواع أخرى تستند إلى SOAP (بروتوكول الوصول إلى الأغراض البسيط Simple Object Access Protocol)، ترتبط كل واجهات برمجة التطبيقات هذه بإنشاء طلبات الخدمة عبر الإنترنت REST هو مفهوم معترف به على نطاق واسع كمنهج لتصميم API القابل للتطوير "عالي الجودة".

من أهم ميزات REST أنها بنية "عديمة الحالة stateless"، هذا يعني أن كل ما يلزم لإكمال الطلب إلى سحابة التخزين مضمن في الطلب بحيث لا يلزم عقد جلسة بين مقدم الطلب وسحابة التخزين، تعتبر REST مهمة للغاية لأن حالة الشبكة لها وقت استجابة غير متوقع والاتصال ليس سريعاً بشكل عام عند مقارنته بشبكة المنطقة المحلية LAN.

REST هو منهج له صلة كبيرة بالطريقة التي يعمل بها الإنترنت. لا تعمل طرق الوصول إلى تخزين الملفات التقليدية التي تستخدم NFS (نظام ملفات الشبكة) أو CIFS (نظام ملفات الإنترنت العام) عبر الإنترنت، بسبب زمن الوصول.

التخزين السحابي مخصص للملفات، والتي يشير إليها البعض كأغراض والبعض الآخر يسميها البيانات غير المنظمة أو غير المهيكلة، مثلاً الملفات المخزنة على الحاسب الشخصي، مثل الصور وجداول البيانات والمستندات، هذه لها تنوع غير عادي وبالتالي فهي غير منظمة.

النوع الآخر من البيانات هو البيانات المنظمة أو الكتلة، على سبيل المثال بيانات قاعدة البيانات، وهي البيانات التي تغذي نظام المعاملات التي تتطلب أداءً معيناً مضموناً أو منخفض التأخير. التخزين السحابي ليس لحالة الاستخدام هذه. يقدر

مركز التصميم الصناعي (IDC) أن ما يقارب من 70% من بيانات الجهاز المخزنة في العالم غير منظمة، وهذا أيضاً هو نوع البيانات الأسرع نمواً. لذا فإن التخزين السحابي هو تخزين للملفات التي يمكن الوصول إليها بسهولة عبر الإنترنت. هذا لا يعني أنه لا يمكنك الوصول إلى التخزين السحابي على شبكة خاصة أو شبكة محلية LAN، والتي قد توفر أيضاً الوصول إلى سحابة التخزين من خلال طرق أخرى، مثل NFS أو CIFS، هذا يعني أن الوصول الأساسي والمفضل يكون بواسطة واجهة برمجة تطبيقات REST.

تعد واجهات برمجة تطبيقات REST لغة محايدة وبالتالي يمكن للمطورين الاستفادة منها بسهولة بالغة باستخدام أي لغة تطوير يختارونها.

قد يتم العمل على الموارد داخل النظام من خلال عنوان URL، لذلك فإن واجهة برمجة التطبيقات ليست "لغة برمجة"، ولكنها طريقة استخدام لغة البرمجة للوصول إلى سحابة التخزين.

تتعلق واجهات برمجة تطبيقات REST أيضاً بتغيير حالة المزود من خلال تمثيل تلك الموارد، ولا يتعلق الأمر باستدعاء أساليب خدمة الويب بالمعنى الوظيفي،

حيث تتمثل الاختلافات الرئيسية بين واجهات برمجة تطبيقات التخزين السحابي المختلفة في عناوين URL التي تحدد الموارد وتنسيق التمثيلات.

أمثلة واجهات برمجة تطبيقات Amazon S3 ، Eucalyptus ، Rackspace
Simple Cloud ، Nivanix APIs ، Mezeo APIs ، Cloud Files APIs
.API

4.8 - الأنظمة الموزعة:

يتم تعريف النظام الموزع على أنه مجموعة من أجهزة الحواسيب المستقلة التي تنظر إلى مستخدميها على أنها نظام واحد متماسك [2]. ويشمل هذا التعريف العديد من الجوانب أهمها:

الجانب الأول هو أن النظام الموزع يحتوي على مكونات مستقلة، والمكونات هنا ليست سوى أنظمة الحاسوب. الجانب الثاني هو أن المستخدمين يعتقدون أنهم يديرون بنظام واحد، هذا يعني أنه بطريقة أو بأخرى، تحتاج أجهزة الحواسيب المستقلة إلى التعاون.

يمكننا تعريف المكونات الأساسية في النظام الموزع على الشكل التالي:

- البرنامج: كود نقوم بكتابته.
- العملية: على ماذا نحصل عندما نقوم بعملية التشغيل.
- الرسالة: تستخدم من أجل عملية الاتصال بين العمليات.
- الحزمة: جزء من الرسالة التي يمكن أن تنتقل.
- البروتوكول: هو وصف رسمي لتنسيقيات الرسائل والقواعد التي يجب على عمليتين اتباعها من أجل تبادل تلك الرسائل.
- الشبكة: هي البنية التحتية التي تربط أجهزة الحاسوب ومحطات العمل والمحطات الطرفية والخدمات وما إلى ذلك و تتكون من أجهزة توجيه متصلة بواسطة روابط اتصال.
- العنصر: يمكن أن تكون عملية أو أي جزء من الأجهزة المطلوبة لتشغيل عملية أو دعم الاتصالات بين العمليات وتخزين البيانات، و غيرها.
- النظام الموزع: هو تطبيق ينفذ مجموعة من البروتوكولات لتنسيق الإجراءات من عمليات متعددة على الشبكة، بحيث تتعاون جميع المكونات معاً لأداء مجموعة واحدة أو صغيرة من المهام ذات الصلة

4.8.1 - ميزات النظام الموزع:

- 1- متسامح مع الخطأ: يبقى النظام قيد التشغيل حتى لو فشلت إحدى مكوناته.
- 2- متوفر: يمكنه استعادة العمليات، والسماح لها باستئناف تقديم الخدمات حتى في حالة فشل بعض المكونات.
- 3- قابل للاسترداد: يمكن للمكونات الفاشلة إعادة تشغيل نفسها والانضمام إلى النظام، بعد إصلاح سبب الفشل.
- 4- الاتساق: يمكن للنظام تنسيق الإجراءات بواسطة مكونات متعددة غالباً في وجود التزامن والفشل. هذا يكمن وراء قدرة النظام الموزع على التصرف كنظام غير موزع.
- 5- قابل للتطوير: يمكن أن يعمل بشكل صحيح حتى مع تغيير حجم بعض جوانب النظام إلى حجم أكبر. على سبيل المثال، قد نقوم بزيادة حجم الشبكة التي يعمل عليها النظام. يؤدي هذا إلى زيادة تواتر انقطاع الشبكة ويمكن أن يؤدي إلى تدهور النظام "غير القابل للتطوير". وبالمثل، قد نقوم بزيادة عدد المستخدمين أو

المخدمات، أو التحميل الكلي على النظام. في نظام قابل للتطوير، لا ينبغي أن يكون لهذا تأثير كبير.

6- الأداء المتوقع: القدرة على توفير الاستجابة المرغوبة في الوقت المناسب.

7- أمن: يصادق النظام على الوصول إلى البيانات والخدمات.

4.8.2 - أهداف النظام الموزع:

ليس الأمر أن الأنظمة الموزعة مبنية فقط لأن الناس لديهم إمكانية بنائها. يحتوي كل نظام موزع على عدد من الأهداف التي يجب تحقيقها أثناء عملية البناء الخاصة منها:

1- جعل الموارد متاحة: من هذه الموارد (الشبكات - صفحات الانترنت -

الملفات - البيانات - مرافق التخزين - أجهزة الكمبيوتر - طابعات).

2- الشفافية: ينطبق مفهوم الشفافية على العديد من جوانب النظام الموزع،

منها (الفشل - التزامن - التمكن من الوصول - الموقع - التكرار - النقل).

3- الانفتاح: فصل استراتيجية العمل عن الآلية.

4- قابلية التوسع: (مشاكل في قابلية التوسع - تقنيات التحجيم).

5- الأخطاء: تشرح الأخطاء التي ارتكبت أثناء تقديم النظام الموزع للمرة

الأولى. تمت صياغة هذه الأخطاء على أنها افتراضات خاطئة وهي كما يلي:

(لديها شبكة موثوقة - الشبكة مؤمنة بالكامل - الشبكة متجانسة- لديها زمن

انتقال صفري - يوجد مسؤول واحد فقط - تكلفة النقل صفر - لديها عرض

النطاق الترددي اللانهائي - الهيكل لا يتغير).

4.8.3 - أنواع الأنظمة الموزعة:

1- نظام الحوسبة الموزعة Distributed computing systems

نوع مهم من الأنظمة الموزعة حيث أنه يُستخدم لمهام الحوسبة عالية الأداء، بشكل

تقريبي يمكن للمرء أن يميز بين مجموعتين فرعيتين:

a. أنظمة الحوسبة العنقودية

b. أنظمة الحوسبة الشبكية

في الحوسبة العنقودية تتكون الأجهزة الأساسية من مجموعة من محطات العمل أو أجهزة الحاسوب المماثلة المتصلة بشكل وثيق عن طريق شبكة محلية عالية السرعة، بالإضافة إلى ذلك تعمل كل عقدة على نفس نظام التشغيل.

يصبح الوضع مختلفاً جداً في حالة الحوسبة الشبكية، تتكون هذه المجموعة الفرعية من أنظمة موزعة يتم إنشاؤها غالباً على شكل اتحاد لأنظمة الحواسيب حيث قد يقع كل نظام ضمن مجال إداري مختلف، وقد يكون مختلفاً تماماً عندما يتعلق الأمر بالأجهزة والبرامج وتكنولوجيا الشبكة.

2- نظم المعلومات الموزعة Distributed information systems

a. أنظمة معالجة التحويلات

b. تكامل تطبيق المؤسسة

فئة مهمة أخرى من الأنظمة الموزعة في المؤسسات التي واجهت ثروة من التطبيقات المتصلة بالشبكة، ولكن تبين أن قابلية التشغيل كانت تجربة سيئة، العديد من حلول البرامج الوسيطة الحالية هي نتيجة العمل مع بنية تحتية كان من الأسهل فيها دمج التطبيقات في نظام معلومات على مستوى المؤسسة.

يتكون التطبيق المرتبط بالشبكة ببساطة من مخدم يقوم بتشغيل هذا التطبيق (غالباً ما يتضمن قاعدة بيانات) وإتاحته للبرامج البعيدة، التي تسمى العملاء أو المستخدمين، يرسل هؤلاء العملاء طلباً إلى المخدم لتنفيذ عملية محددة وبعد ذلك يتم إرسال استجابة مرة أخرى. من أجل العمل عند أدنى مستوى للعملاء يتم تغليف عدد من الطلبات، ربما لمخدمات مختلفة في طلب واحد أكبر وتنفيذها كعمالة موزعة و هذا ما يسمى بالتكامل. الفكرة الأساسية هي أنه يتم تنفيذ جميع الطلبات أو عدم تنفيذها. نظراً لأن التطبيقات أصبحت أكثر تعقيداً وتم فصلها تدريجياً إلى مكونات مستقلة (لا سيما التمييز بين مكونات قاعدة البيانات ومكونات المعالجة)، أصبح من الواضح أن التكامل يجب أن يتم أيضاً عن طريق السماح للتطبيقات بالاتصال ببعضها البعض مباشرة. وقد أدى هذا الآن إلى صناعة ضخمة تركز على تكامل تطبيقات المؤسسة (EAI).

3- الأنظمة المنتشرة الموزعة Distributed pervasive systems

a. أنظمة منزلية

b. أنظمة الرعاية الصحية الإلكترونية

c. شبكات الاستشعار

تتميز الأنظمة الموزعة التي تمت مناقشتها حتى الآن إلى حد كبير باستقرارها حيث العقد ثابتة ولديها اتصال دائم وعالي الجودة إلى حد ما بالشبكة، نوعاً ما، يتحقق هذا الاستقرار نوعاً ما من خلال التقنيات المختلفة لتحقيق شفافية التوزيع، على سبيل المثال هناك العديد من الطرق التي يمكننا من خلالها خلق الوهم بأن المكونات في بعض الأحيان فقط قد تفشل، وبالمثل هناك جميع أنواع الوسائل لإخفاء موقع الشبكة الفعلي للعقدة، مما يسمح للمستخدمين والتطبيقات بشكل فعال بالاعتقاد بأن العقد تظل في مكانها، ومع ذلك فقد تغيرت الأمور منذ إدخال أجهزة الحوسبة المحمولة والمدمجة، مما أدى إلى ما يشار إليه عموماً بالأنظمة المنتشرة، كما يوحي اسمها تهدف الأنظمة المنتشرة إلى الاندماج بشكل طبيعي في بيئتنا، هي أيضاً أنظمة موزعة بشكل طبيعي، و ما يجعلها فريدة بالمقارنة مع أنظمة الحوسبة والمعلومات الموصوفة حتى الآن هو أن الفصل بين المستخدمين ومكونات النظام أكثر غموضاً، غالباً لا توجد واجهة واحدة مخصصة مثل مجموعة الشاشة / لوحة المفاتيح، بدلاً من ذلك غالباً ما يكون النظام واسع الانتشار مزوداً بالعديد من المستشعرات التي تلتقط جوانب مختلفة من سلوك المستخدم، وبالمثل قد يكون لديها عدد لا يحصى من المشغلات لتقديم المعلومات والتعليقات، وغالباً ما تهدف بشكل مقصود إلى توجيه السلوك.

تتميز العديد من الأجهزة في الأنظمة المنتشرة بأنها صغيرة ومزودة بالبطارية ومنتقلة ولا تحتوي إلا على اتصال لاسلكي، على الرغم من عدم تطبيق كل هذه الخصائص على جميع الأجهزة، هذه ليست بالضرورة خصائص مقيدة كما هو الحال في الهواتف الذكية، ومع ذلك فإن حقيقة أننا غالباً ما نحتاج إلى التعامل مع تعقيدات الاتصالات اللاسلكية والمنتقلة تتطلب حلاً خاصة لجعل النظام المنتشر شفافاً أو غير مزعج قدر الإمكان.

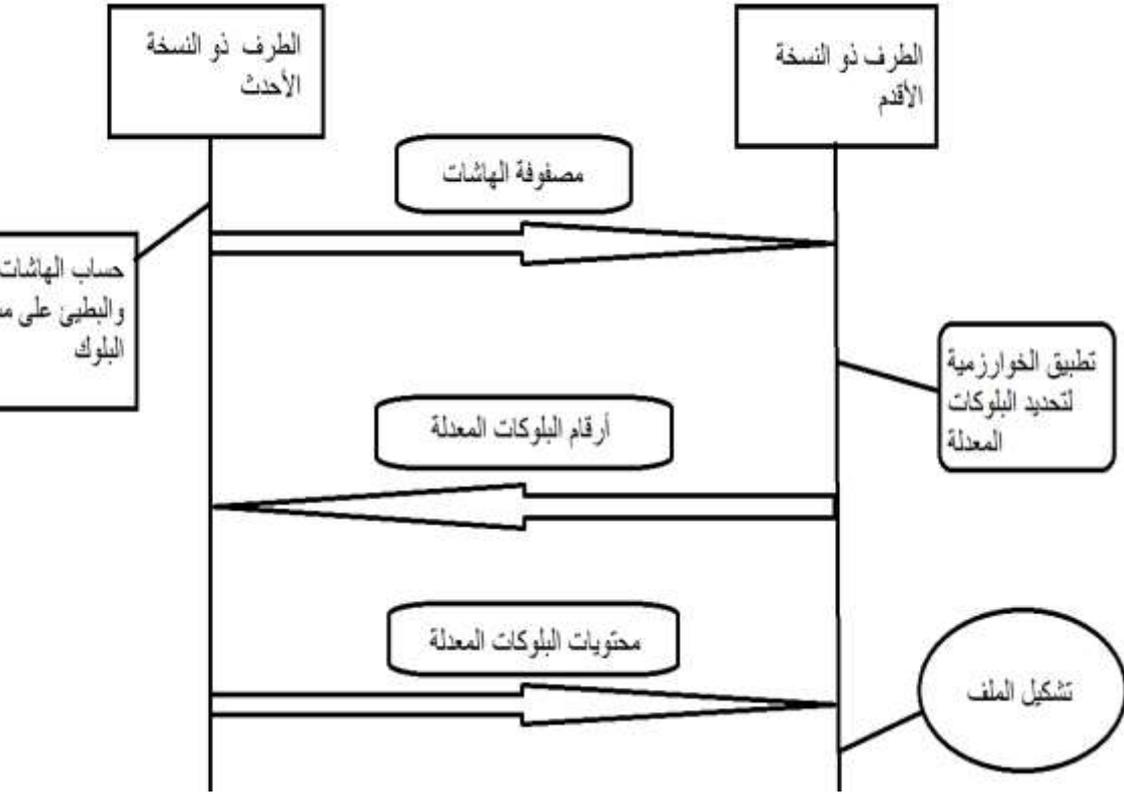
5 - النتائج والمناقشة:

5.1 - مقدمة:

قمنا في هذا البحث بالعمل على تطوير خوارزمية تعمل على مزامنة (تحديث) ملف أو مجموعة من الملفات بين طرفين حيث أحد الطرفين يحوي نسخة أحدث من النسخة المتوفرة عند الطرف الآخر و نعتد على تاريخ التعديل في تحديد النسخة الأحدث لتبدئ الخوارزمية في التنفيذ ونحصل في النهاية على نسخة متطابقة من الملف عند الطرفين بعد تبادل عدد معين من الرسائل التي تدل على مواقع التعديل وتبادل محتوى البلوكات المعدلة فقط.

5.2 - توصيف الخوارزمية:

يمكننا توصيف الخوارزمية المقترحة بالشكل التالي:



وتتلخص خطوات الخوارزمية المقترحة بالبنود التالية:

✓ يقوم أحد الطرفين بإرسال تاريخ آخر تعديل للملف الى الطرف الأخر

ليقوم بمقارنة التاريخين وتحديد الطرف المالك للنسخة الأحدث.

✓ يقوم الطرف المالك للنسخة الأحدث بحساب الهاشات (hashes) على

مستوى البلوك وتشكيل مصفوفتين:

➤ مصفوفة الهاشات السريعة حجمها n هاش بحيث قمنا باختبار هاش سريع الحساب وهو أخذ أول ثلاث بايتات من بداية البلوك وآخر ثلاث بايتات من نهاية البلوك لتشكيل هاش سريع الحساب (كثير التوافق) بحجم 6 بايت.

➤ مصفوفة الهاشات البطيئة بحجم n هاش وبذلك اعتمدنا على خوارزمية MD5 المعروفة لحسابه حيث يتميز بصعوبة في الحساب ولكن صعب جداً وجود تطابق بين بلوكين في حال اختلاف البيانات بينهما.

✓ يقوم بعدها بإرسال المصفوفتين الى الطرف الآخر ذو النسخة الأقدم.
✓ يقوم الطرف ذو النسخة الأقدم وهو مسؤول عن تطبيق آلية للبحث عن البلوكات وتحديد البلوكات المعدلة وموضعها وذلك من خلال توليد مصفوفة هاشات سريعة على مستوى الباييت ومن اجل كل هاش سريع تم استقباله يعمل على البحث عنه ويكون امام حالتين:

➤ حالة وجود تطابق مع أحد الهاشات السريعة لديه ليقوم بتوليد الهاش البطيء ومقارنته مع الهاش البطيء المستقبلي وبالتالي أيضا يكون هنا أمام خيارين:

❖ في حال تطابق الهاشات البطيئة معاً: يتأكد أن محتويات

البلوك موجودة لديه.

❖ في حال عدم التطابق بين الهاشات البطيئة: يقوم بإكمال

البحث عن هاش سريع اخر متطابق مع الهاش السريع

المستقبل.

➤ في حال عدم حدوث تطابق ابداً مع أية هاش سريع عندها

أن البلوك تم تعديله ويسجل رقمه ليتم طلبه لاحقاً من الطرف ذو

النسخة الاحدث.

✓ بعد العملية السابقة نكون قد حصلنا على أرقام البلوكات المعدلة وأرقام

البلوكات غير المعدلة.

✓ يقوم عندها بإرسال أرقام البلوكات المعدلة الى الطرف ذو النسخة الأحدث

طالباً منه محتويات البلوكات.

✓ يتم ارسال محتويات البلوكات المعدلة من الطرف ذو النسخة الأحدث الى

الطرف الأخر.

✓ في النهاية: يعمل الطرف ذو النسخة الأحدث بإعادة تشكيل الملف من

البلوكات الموجودة لديه إضافة الى البلوكات المستقبلية لنحصل أخيراً على

نسخة متطابقة من الملف عند الطرفين (تمت عملية المزامنة بنجاح دون الحاجة لإرسال كامل محتوى الملف والاستعاضة عنها بإرسال أجزاء محددة معدلة من الملف).

5.3 - معايير تقييم أداء عمل الخوارزمية:

تم تقييم عمل وأداء الخوارزمية المقترحة بناءً على عدد من المتغيرات:

✓ استخدام احجام ملفات متنوعة (100KB-1MB-10MB...).

✓ استخدام تعديلات متنوعة على محتوى الملف (عشوائي - مواضع

محددة).

✓ تعديل حجم البلوك ودراسة تأثيره على عمل الخوارزمية (1KB-10KB-

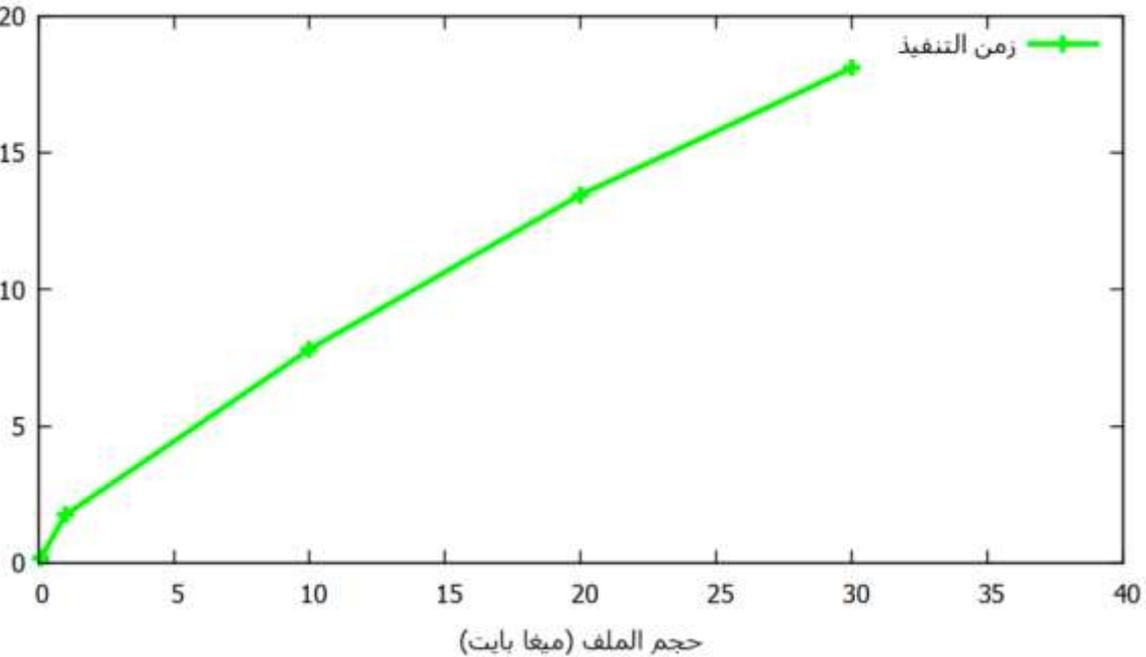
100KB-1MB...).

5.4 - النتائج العملية:

5.4.1 - تقييم عمل الخوارزمية بناءً على أحجام ملفات متنوعة:

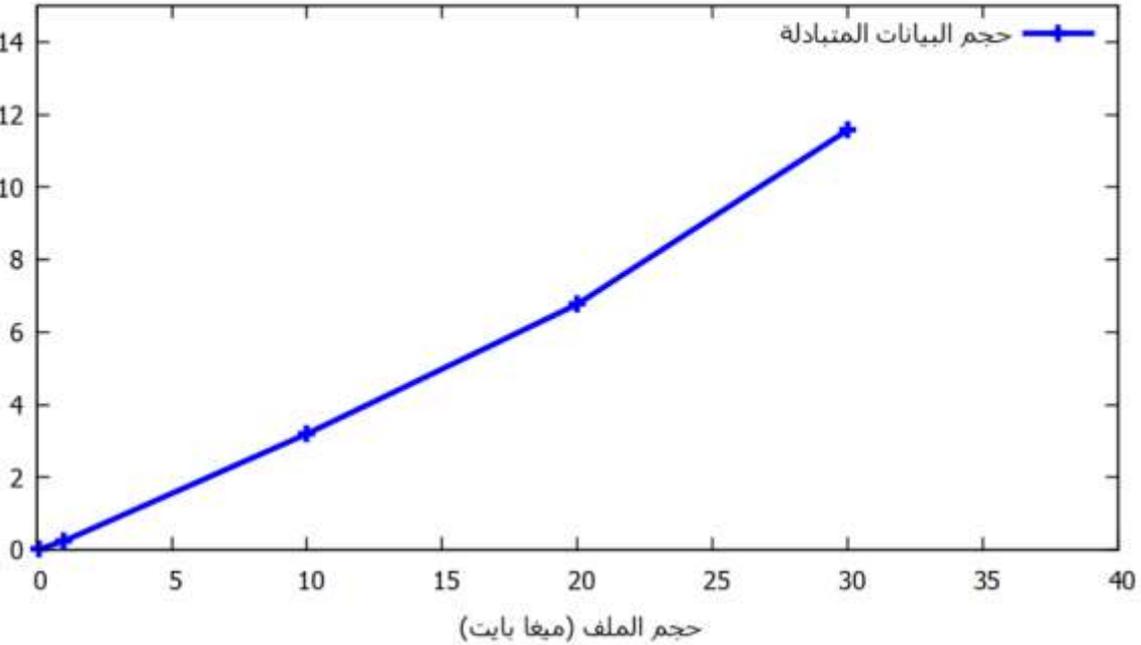
تم بتثبيت حجم البلوك وحجم التعديلات والعمل على استخدام أحجام ملفات مختلفة ومقارنة زمن تنفيذ الخوارزمية المقترحة كما في الشكل (1) لنلاحظ أن زمن تنفيذها يزداد مع ازدياد حجم الملف المراد مزامنته ولكن يبقى حجم البيانات المتبادلة على الشبكة اقل بشكل ملحوظ مقارنة مع ارسال كامل الملف كما في الشكل (2).

زمن تنفيذ الخوارزمية المقترحة بناءً على أحجام ملفات مختلفة



الشكل (1) زمن تنفيذ الخوارزمية المقترحة بناءً على أحجام ملفات مختلفة

حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام ملفات مختلفة



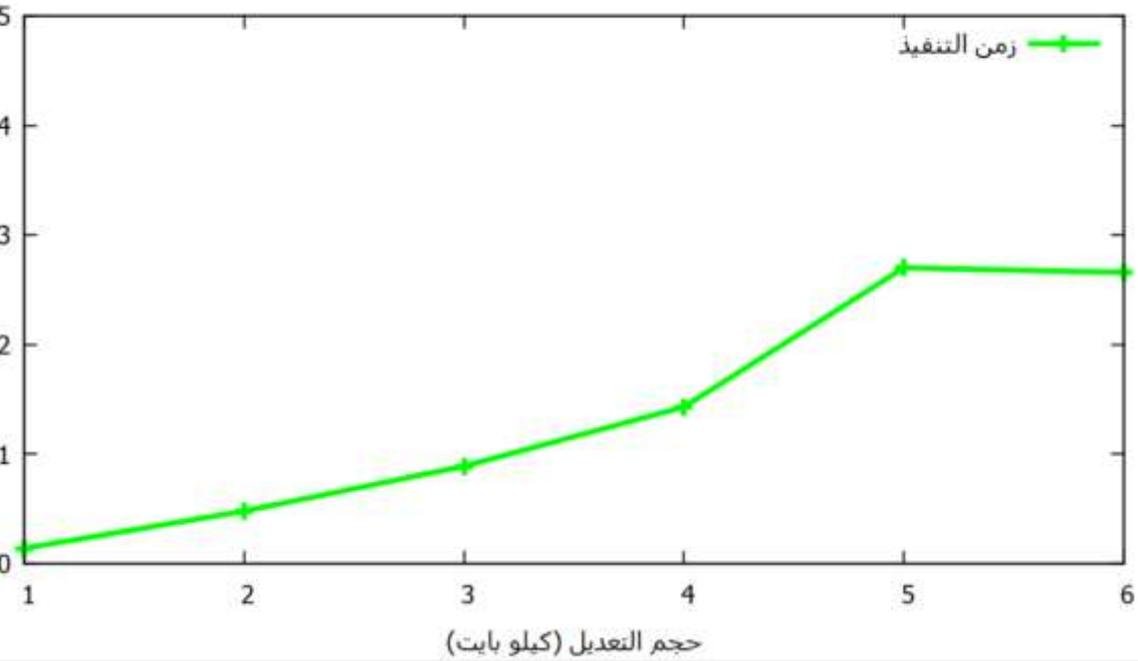
الشكل (2) حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام ملفات مختلفة

5.4.2 - تقييم عمل الخوارزمية المقترحة بناءً على تعديلات مختلفة في

محتوى الملف :

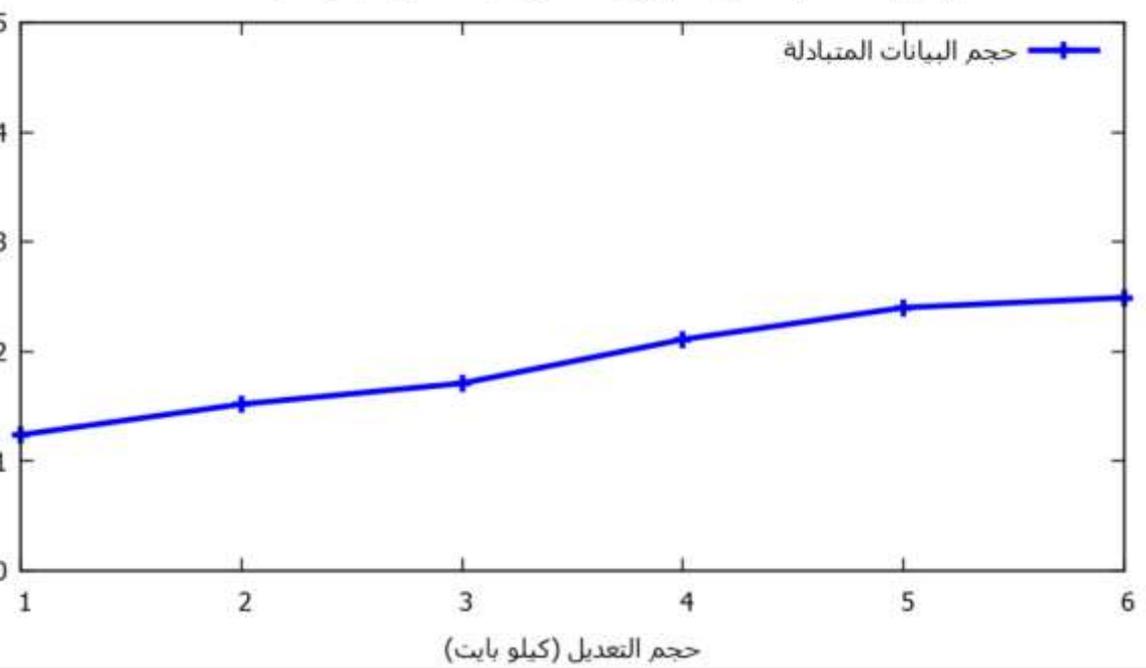
تم الحفاظ على حجم الملف وحجم البلوك ثابت وتعديل حجم التعديلات (الفروق) الحاصلة في الملف ودراسة اداء الخوارزمية حيث نلاحظ من الشكل (3) زيادة في زمن تنفيذ الخوارزمية مع زيادة حجم التعديلات في محتوى الملف بسبب الحاجة الى مزيد من عمليات توليد الهاش الصعب (hard) ولكن تبقى الخوارزمية تقدم أداء أفضل من الآليات التقليدية المعتمدة على إرسال كامل الملف كما في الشكل (4).

زمن تنفيذ الخوارزمية المقترحة بناءً على أحجام تعديلات مختلفة



الشكل (3) زمن تنفيذ الخوارزمية المقترحة بناءً على أحجام تعديلات مختلفة

حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام تعديلات مختلفة



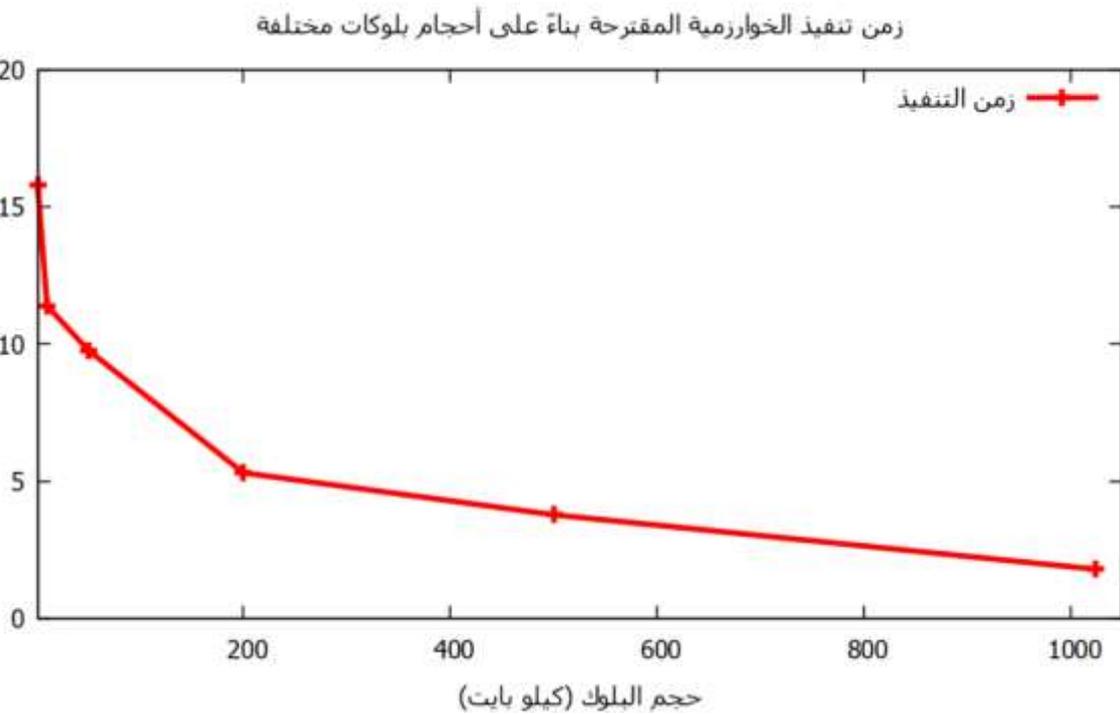
الشكل (4) حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام

تعديلات مختلفة

5.4.3 - تقييم عمل الخوارزمية المقترحة بناءً على أحجام بلوكات متنوعة:

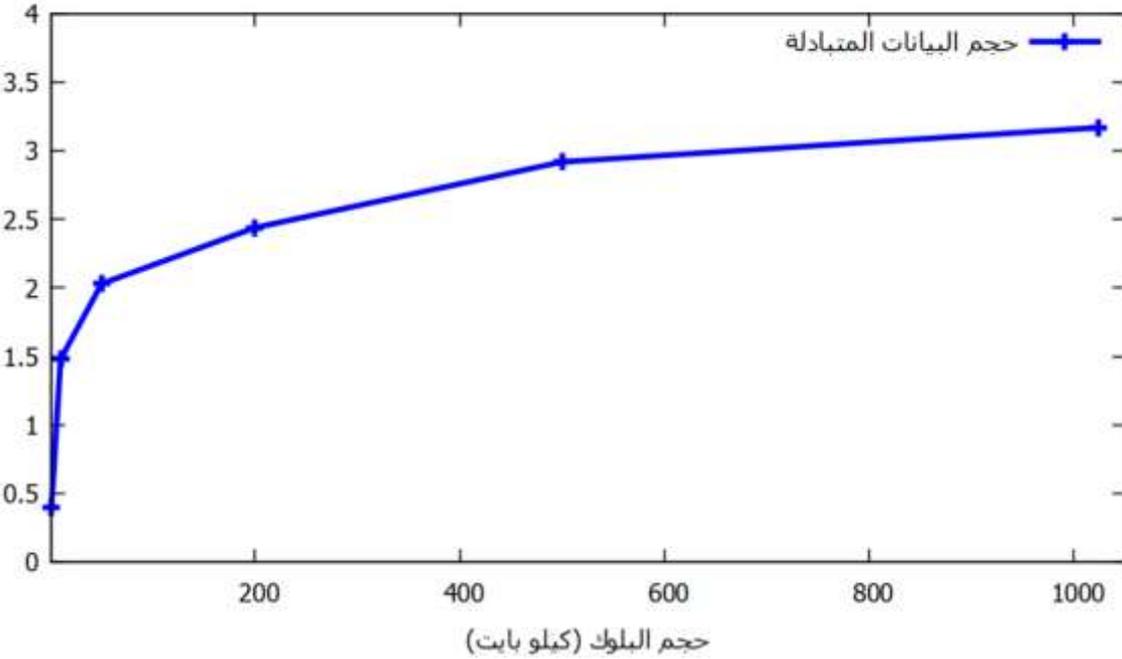
تم تثبيت حجم الملف وحجم التعديلات في محتواه والعمل على استخدام أحجام بلوكات مختلفة لنلاحظ من الشكل (5) أن زمن تنفيذ الخوارزمية المقترحة ينقص مع زيادة حجم البلوك الذي بدوره يقلل من عمليات توليد الهاشات وتقليل في زمن

البحث عن تموضع البلوكات ولكن هذا النقصان في حجم البلوك يؤدي الى زيادة في حجم البيانات المتبادلة بشكل ملحوظ كون الحاجة الى ارسال بلوكات بأحجام أكبر عبر الشبكة كما في الشكل (6).



الشكل (5) زمن تنفيذ الخوارزمية المقترحة بناءً على أحجام بلوكات مختلفة

حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام بلوكات مختلفة



الشكل (6) حجم البيانات المتبادلة في الخوارزمية المقترحة بناءً على أحجام

بلوكات مختلفة

5.4.4 - تقييم عمل الخوارزمية المقترحة بناءً على مواضع تعديلات مختلفة :

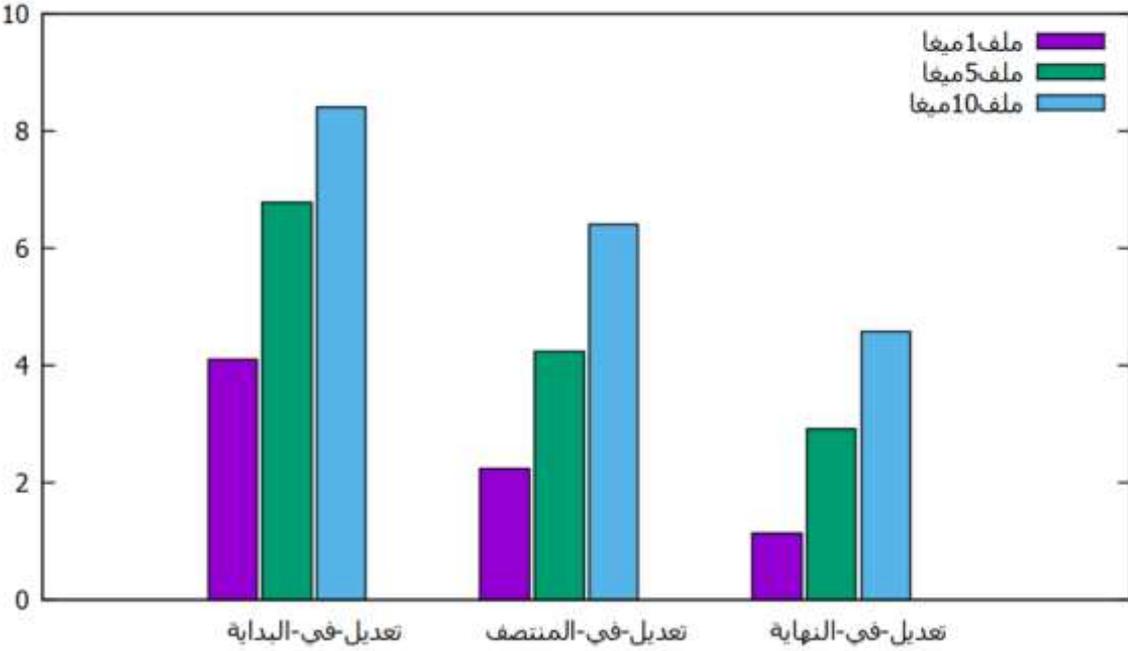
تم هنا العمل على ثلاث أحجام مختلفة من الملفات والقيام بتعديلات في محتوى

الملف قبل عملية المزامنة في ثلاث مواضع مختلفة (البداية - المنتصف -

النهاية) لنجد أن زمن التنفيذ يزداد مع اقتراب الاختلاف في محتوى الملف من

بدايته ولكن تبقى الخوارزمية تقدم أداء جديد جداً مقارنة مع عملية إرسال كامل الملف على الشبكة كما في الشكل (6).

زمن تنفيذ الخوارزمية المقترحة بناءً على مواضع تعديلات مختلفة



الشكل (6) زمن تنفيذ الخوارزمية المقترحة بناءً على مواضع تعديلات مختلفة

6 - الاستنتاجات والتوصيات:

بناءً على ما سبق يمكننا تلخص مجموعة من الاستنتاجات والتوصيات:

❖ تعتبر عملية ارسال كامل الملف المتبعة في أنظمة التخزين السحابي

الحالية غير فعالة وذلك عند الملفات ذات الأحجام الكبيرة كونها تستهلك

عرض حزمة كبير وبالتالي كلفة عالية.

❖ قللت الخوارزمية المقترحة من حجم الرسائل المتبادلة بين طرفي المزامنة

بشكل ملحوظ مقارنة مع أنظمة ارسال كامل الملف وبالتالي تقليل في

حجم استهلاك عرض الحزمة المتوفر.

❖ أثبتت الخوارزمية المقترحة فعاليتها في الحفاظ على أداء جيد جدا مع

زيادة حجم الملف بين الطرفين.

❖ مع زيادة حجم التعديلات في الملف المراد مزامنته تبقى الخوارزمية قادرة

على العمل مع زيادة نسبية في حجم الحسابات عند الطرف الأقدم.

❖ زيادة حجم البلوك في الخوارزمية المقترحة يؤدي الى تقليل في عدد

الحسابات اللازمة ولكن زيادة في حجم البيانات المرسله.

❖ من الممكن تطوير عمل الخوارزمية لتكون قادرة على الاستجابة لأنواع

أخرى من التعديلات التي ممكن حدوثها في الملف (إضافة - حذف).

❖ من الممكن الاستفادة من عمل الخوارزمية للعمل على مزامنة الملفات بين

أكثر من طرفين.

7 - الخاتمة:

تم في هذا البحث العمل على تطوير خوارزمية

8 - المراجع:

1. D.Ford, et al. “**Availability in Globally Distributed Storage Systems**”, Dept. of Industrial Engineering and Operations Research Columbia University, 2010.
2. M.V.Steen and A.S.Tanenbaum, “**A Brief Introduction to Distributed Systems**”, Springer, 2016.
3. C.Yan , “**Cloud Storage Services**”, CENTRIA UNIVERSITY OF APPLIED SCIENCES, Information Technology, 2017.
4. R.A.Rajan and S.Shanmugapriyaa, “**Evolution of Cloud Storage as Cloud Computing Infrastructure Service**”, IOSR Journal of Computer Engineering (IOSRJCE), (May–June 2012), PP 38–45.
5. L.Posani, A.Paccoia, M.Moschettini, “The carbon footprint of distributed cloud storage”, June 26, 2019.
6. S.L.Obrutsky, “**Cloud Storage: Advantages, Disadvantages and Enterprise Solutions for Business**”, Eastern Institute of Technology, Hawke’s Bay, New Zealand, 2016.

7. X.Gao,et al. , **“Building a Distributed Block Storage System for Cloud Infrastructure”**, Conference Paper, January 2011.

نمذجة و تحليل أداء شبكات برتوكول MODBUS

التسلسلي

طالب الدكتوراه: رضوان صبحي المحمد

كلية الهمك - جامعة البعث

اشراف الدكتور: مسعود الأتاسي

الملخص:

تعتبر الشبكات الصناعية التي تستخدم بروتوكول MODBUS من أكثر الأنواع انتشاراً و ذلك نتيجة المزايا التي يدعمها هذا البرتوكول و التي تلبي متطلبات الأتمتة الصناعية، و ظهرت العديد من اصدارات بروتوكول MODBUS لتلبي التطور المتسارع في منظومات الشبكات الصناعية.

لدراسة سلوك بروتوكول MODBUS و تحليل أداء الشبكات الصناعية التي يعمل عليها لا بدّ من وضع نماذج تحاكي سلوك البرتوكول و تدعم إجراء عمليات الاختبار و التحليل بحيث تتيح دراسة العوامل المؤثرة في بارامترات الأداء و التنبؤ بسلوك الشبكة المستقبلي في حال الرغبة بتطوير الشبكة الصناعية التي يعمل عليها.

في هذا البحث تمّ وضع نموذج لشبكة صناعية تستخدم بروتوكول MODBUS التسلسلي باستخدام محاكي الشبكات OPNET، و بعد نمذجة جميع الاعتبارات التي يدعمها هذا البرتوكول تمت دراسة مجموعة من بارامترات الأداء مثل (زمن الاستجابة، و تأخير الانتقال للرسائل، و استخدامية الشبكة) بهدف تحديد أفضل أداء لعمل الشبكة و تحقيق متطلبات الزمن الحقيقي.

الكلمات المفتاحية: بروتوكول MODBUS ، تحليل أداء الشبكات الصناعية، العوامل

المؤثرة في أداء الشبكات، تحليل الأداء باستخدام OPNET.

Modeling And Analysis the Performance of Serial MODBUS Network

Abstract:

Industrial networks that use the MODBUS protocol are considered one of the most common types, because of the advantages supported by this protocol and that meet the requirements of industrial automation, and many versions of the MODBUS protocol have appeared to meet the rapid development in the industrial network system.

To study the behavior of the MODBUS protocol and analyze the performance of the industrial networks on which it operates, it is necessary to develop models that simulate the protocol and support the conduct of analyzes so as to allow studying the factors affecting performance parameters and predicting the future behavior of the network in the event that it is desired to develop the industrial network on which it operates.

In this research, an industrial network using MODBUS protocol was modeled using the OPNET network simulator, and all considerations supported by this protocol were modeled, and then a set of performance parameters such as (response time, delay end to end, utilization) were studied. In order to determine the best performance of the network and achieve the requirements of real time.

The Key words: MODBUS protocol, industrial network performance, performance analysis using OPNET.

1- المقدمة:

يُعتبر بروتوكول MODBUS التسلسلي بروتوكول اتصال رقمي ثنائي الاتجاه يدير الإرسال بين الأجهزة الصناعية الذكية، ويُعد شبكة محلية مخصصة للأتمتة المعامل. [1] في شبكات MODBUS ذات الإرسال ثنائي الاتجاه من الممكن قراءة البيانات من و إلى الحساسات و المشغلات، و ينتج عن هذا النوع من الاتصال ثنائي الاتجاه للناقل توفيراً كبيراً بعدد الكابلات المُستخدم و بتالي خفض تكاليف الإنتاج، و يجب أن يحتوي كل جهاز يُراد ربطه بالناقل على واجهة للاتصال يتم من خلالها ضبط إعدادات الاتصال بالناقل. [1]

إن أداء شبكات MODBUS يتأثر بمجموعة من البارامترات المختلفة لذا لا بدّ من وضع نماذج تحاكي سلوك البروتوكول باستخدام أحد برامج المحاكاة الشبكية لدراسة و تحليل العوامل المؤثرة فيها. [2]

يدعم محاكي الشبكات OPNET نمذجة سلوك بروتوكولات الاتصال و يتيح توفير بيئة مفتوحة المصدر لنمذجة كافة الاعتبارات التي يدعمها أي نوع من بروتوكولات الاتصال، و تمّ الاستفادة من هذه المزايا لنمذجة شبكة صناعية تستخدم ناقل MODBUS التسلسلي و تحليل بارامترات مؤثرة في الأداء لمعرفة سلوك الشبكة المستقبلي بهدف تحسينها و المحافظة على موثوقيتها. [3]

2- مشكلة البحث:

- ❖ صعوبة إجراء عمليات التنبؤ و الاختبار على شبكة صناعية واقعية.
- ❖ الصعوبة في توسعة و تطوير الشبكة دون التأثير على الأداء العام لها.
- ❖ عدم القدرة على تحديد أهم العوامل المؤثرة في أداء الشبكات الصناعية أثناء عملها.
- ❖ مشكلة مادية كبيرة عند عدم تلبية الشبكة لمتطلبات العمل في الأداء المطلوب.

3- هدف البحث:

إنّ الهدف الرئيسي هو بناء نموذج يحاكي سلوك شبكة صناعية تعمل على بروتوكول MODBUS التسلسلي آخذين بعين الاعتبار جميع الخصائص التي يدعمها و من ثم إجراء عمليات التحليل و الاختبار بهدف:

- ✓ تحديد أهم العوامل المؤثرة في بارامترات الأداء.
- ✓ المقارنة بين بارامترات الأداء عند كل سيناريو اختبائي لتحديد القيم الأفضل.
- ✓ إجراء عمليات التطوير على النموذج قبل تنفيذها على الشبكة الواقعية.
- ✓ دراسة التكلفة المادية و متطلبات التوسعة للشبكة من خلال النموذج المُصمّم.
- ✓ القدرة على التنبؤ بسلوك الشبكة المستقبلي عند إضافة معدات جديدة على الشبكة.

4- الدراسات المرجعية:

اهتم العديد من الباحثين حول العالم بمجال نمذجة الشبكات الصناعية و تحليل أدائها و بالأخص التي تستخدم بروتوكول MODBUS نظراً لانتشاره الواسع و سنذكر بعض الأبحاث التي تمّ الوقوف عندها و النظر في نتائجها:

- اقترح الباحثان Yao Yuanyuan, Chen Meng خوارزمية محسّنة لطول إطار الاتصال التكميلي استناداً على بروتوكول Modbus لتعديل طول الإطار بهدف تحديد حجم الإطار الأفضل للإرسال وفقاً لمتوسط معدل خطأ الإطار في الفترة الزمنية، و تمّ استخدام طريقة "تقليل سريع ، زيادة بطيئة" لضبط طول إطار البيانات عند مستويات مختلفة من FER (Frame Error Rate) معدل خطأ الإطار، لا تعمل هذه الخوارزمية على تحسين معدل الإرسال فحسب بل أيضاً على تحسين استقرار الاتصال و بتالي الأداء. [4]

- الورقة البحثية [5] درست العديد من المعايير لتقييم أداء أجهزة الشبكات من نوع Modbus، و تتضمن: (1) وقت الاستجابة لطلبات Modbus، (2) الحد الأقصى لعدد الطلبات التي يمكن التعامل معها بنجاح بواسطة أجهزة Modbus في فترة زمنية

مُحدّدة، و3) مراقبة أجهزة Modbus عند تعرضها لهجوم رفض مُوزّع للخدمة (Distributed Denial of Service)، و تمّ استخدام دارتين الكترونيّتين ذات تكلفة منخفضة و هي (ESP8266 و Raspberry Pi / OpenPLC) لتقييم أداء بروتوكول Modbus.

- الهدف من المقال [6] هو تقييم أداء ناقل صناعي وآليات تنفيذ دورة الرسائل حيث قدّم نموذجاً يستند على شبكة بتري الملونة لإجرائيات الإرسال و تنفيذ دورة الرسائل (MAC)) (Medium Access Control) لطبقة ربط المعطيات للبروتوكولات Fieldbus، وأكدت النتائج على الحاجة لتحديد بارامتر زمن التشغيل وهو معكوس معدل الإرسال (transmission rate) بشكل مناسب من أجل ضمان الأداء الأمثل لشبكة Fieldbus.

- قام الباحثان Beata Krupanek and Ryszard Bogacz بإجراء بحث [7] باستخدام المحاكي OPNET لدراسة أداء الأنظمة اللاسلكية متعددة العقد، وتمّ من خلال هذا البحث نمذجة شبكة مؤلفة من عدة عقد لاسلكية باستخدام برنامج OPNET و دراسة جودة الخدمة و الأداء للشبكة من حيث تلبيتها لمُحدّدات الزمن الحقيقي و دراسة التأخيرات في عملية نقل البيانات.

- في الدراسة المرجعيّة [8] قدّم الباحثون تحليلاً لأداء بروتوكول الاتصال Modbus الذي يتمّ تنفيذه باستخدام محاكي الشبكات (NS-3)، و يُركّز البحث على تقييم الأداء من خلال زمن الاستجابة مرتبطاً بعدد العقد والطوبولوجيا، و أظهرت النتائج أن طول الحزمة ليس له تأثيراً كبيراً على وقت الاستجابة في كلا النوعي للطوبولوجيا (الناقل و النجمي) و بتالي ليس له أثراً كبيراً على الأداء.

- في البحث [9] ناقش الباحثون أداة لتحليل وقت الاستجابة والجدولة لاتصالات بروتوكول Modbus عبر ناقل RS-485، حيث يتمّ جمع أزمنة استجابة مجموعة الرسائل بواسطة جهاز Modbus مُتخصّص ثم إرسالها إلى البرنامج حيث يتمّ التحليل، ثم يتمّ نشر تطبيق Modbus عبر شبكة RS-485 بشكل يتناسب مع الأداء.

- قدم الباحثون Jia Hao, Jiechang Wu, Chaoyou Guo بحثاً [10] تمّ من خلاله نمذجة و تحليل أداء بروتوكول CAN ، حيث تمّ بالبداية نمذجة الناقل بشكل هرمي انطلاقاً من نموذج الشبكة و من ثم نموذج العقدة و من ثم نموذج العمليّة مع الأخذ بعين

الاعتبار كلاً من وظائف معالجة الأخطاء و تحسس حالة القناة و أولوية العقد لتفادي التصادم، و بعد الانتهاء من بناء النموذج تم دراسة بعض العوامل المؤثرة في الأداء مثل أولوية العقدة و حجم الإطار و الأزمنة الفاصلة بين عمليات الإرسال.

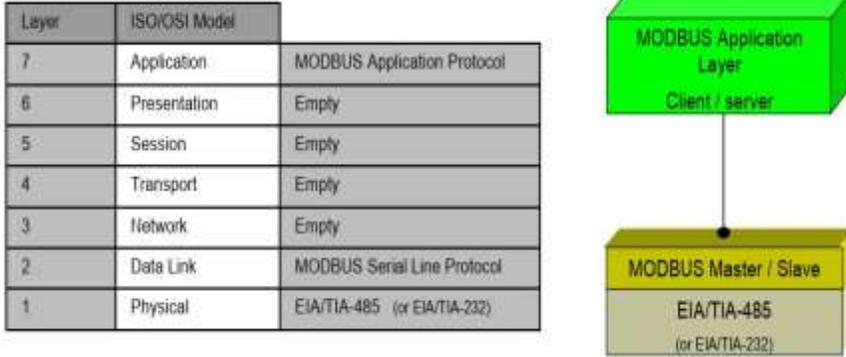
- تم في البحث [11] دراسة بعض العوامل المؤثرة في تصميم الشبكات الصناعية Fieldbus من خلال المحاكاة باستخدام برنامج Opnet و أثر هذه العوامل على أداء الشبكات، و تم اختيار ثلاث عوامل وهي معدل الإرسال (إما 2.5Mbps أو 1.5Mbps) و طبولوجيا الشبكة (إما حلقي أو نجمي) و نوع مجمع الشبكة المركزي (إما hub أو switch) مما يقود إلى اختبار ثمانية سيناريوهات (2^3) في المحاكاة OPNET، و تم في كل سيناريو تشغيلي حساب مجموع عدد الرزم المرسل بالمقارنة مع عدد الرزم المستقبلية و تقسيم المجموعين للحصول على نسب تعكس أداء كل حالة اختبار.

- قام الباحث العراقي قتيبة علي بنشر مقال [12] لدراسة و تحليل أداء شبكات إترنت الصناعية باستخدام برنامج المحاكاة أوبنيت ، ففي البداية تم التأكد من قابلية برنامج أوبنيت في نمذجة الشبكات الصناعية من خلال مقارنة نتائج تجارب شبكة عملية مع نتائج المحاكاة و من ثم تم دراسة تأثير عدد من العوامل على الأداء مثل طول الحزمة و عدد العقد و معدل إنتاجية العقد و حمل FTP موجه لعقدة محددة، وأظهرت النتائج أنه يمكن استخدام برنامج OPNET بكفاءة لمحاكاة الشبكات الصناعية.

5- بروتوكول MODBUS التسلسلي:

هو بروتوكول Fieldbus ذو نظام ارسال (Master/Slave)، فالعقدة "الرئيسية" تُصدر أوامر صريحة إلى إحدى العقد "التابعة" وتقوم بمعالجة الاستجابات، ولا تقوم العقد التابعة عادةً بنقل البيانات دون طلب من العقدة الرئيسيّة، ولا تتواصل مع العقد التابعة الأخرى إلا عن طريقها. [2]

يُعطي الشكل (1) تمثيلاً عاماً لطبقات بروتوكول MODBUS مقارنة بالطبقات السبع لنموذج (OSI) حيث يُلاحظ وجود ثلاث طبقات و هي (الطبقة الفيزيائية و طبقة ربط المعطيات و طبقة التطبيق). [2]



الشكل (1): طبقات بروتوكول MODBUS مقارنة بطبقات (OSI).

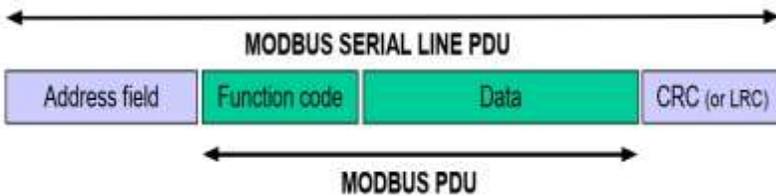
1-5 إطار بروتوكول MODBUS التسلسلي:

يُحدد بروتوكول MODBUS وحدة بيانات بروتوكول بسيطة (PDU) مستقلة عن طبقات الاتصال الأساسية، و تتألف من حقل البيانات و حقل الوظيفة كما يظهر في الشكل(2).



الشكل (2): وحدة البيانات في بروتوكول MODBUS.

يقوم بروتوكول MODBUS على الناقل بإضافة بعض الحقول على وحدة بيانات البروتوكول، ويُظهر الشكل (3) إطار MODBUS التسلسلي المُرسَل من العقدة الرئيسية.



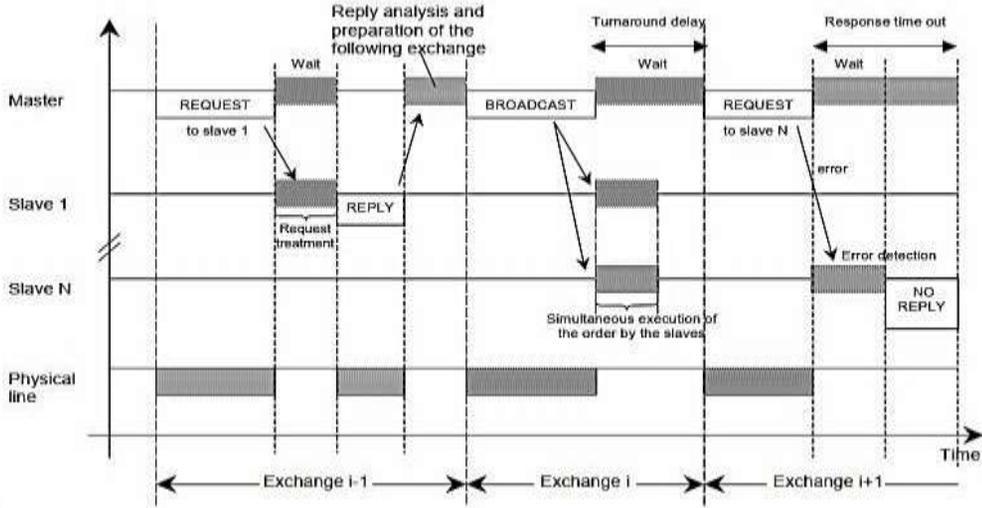
الشكل (3): إطار MODBUS التسلسلي.

يتألف مما يلي:

- حقل العنوان: يحتوي على عنوان الجهاز التابع فقط، و تقع عناوين العقد التابعة في نطاق العشري (0-247)، و يُخاطب الجهاز الرئيسي الجهاز التابع من خلال وضع عنوانه في حقل عنوان الرسالة، وعندما يُرجع الجهاز التابع استجابته فإنه يُكرّر وضع عنوانه في حقل العنوان للاستجابة للسماح للجهاز الرئيسي معرفة وصول البيانات إلى عنوانها المطلوب و بشكل صحيح.
- حقل الوظيفة: يشير إلى رمز الوظيفة للجهاز التابع أي نوع الإجراء الذي يجب القيام به.
- حقل تدقيق الأخطاء: هو نتيجة حساب "التحقق من التكرار" الذي يتم إجراؤه على محتويات الرسالة، يتم استخدام نوعين من طرق الحساب اعتماداً على نمط الإرسال المستخدم (RTU أو ASCII).

5-2 المخطط الزمني لبروتوكول MODBUS التسلسلي:

يوضح الشكل (4) الرسم البياني الزمني لثلاثة سيناريوهات نموذجية لاتصالات (Master / Slave). [2].



الشكل (4): المخطط الزمني للإرسال (Master / Slave) لناقل MODBUS.

السيناريو الأول: يقوم فيه الجهاز الرئيسي "Master" بإرسال طلب "REQUEST" للعقدة التابعة "Slave 1" و التي تستجيب برسال "REPLY" بعد فترة زمنية محددة "REQUEST TREATMENT" و يكون الناقل "Physical Line" خلال عملية ارسال كلاً من "REQUEST" و "REPLY" مشغولاً كما هو موضح بالشكل (4).

السيناريو الثاني: يقوم فيه الجهاز الرئيسي "Master" بإرسال طلب عام "BROADCAST" لجميع الأجهزة التابعة "Slaves" و التي تستجيب فيها للطلب المرسل دون إرسال رد و يكون الناقل "Physical Line" خلال عملية ارسال "BROADCAST" مشغولاً كما هو موضح بالشكل (4).

السيناريو الثالث: يقوم فيه الجهاز الرئيسي "Master" بإرسال طلب "REQUEST" للعقدة التابعة "Slave N" و يعاني هذا الطلب من حدوث خطأ تقوم العقدة التابعة باكتشافه "Error Detection" فلا تقوم بإرسال طلب إجابة و عندما يتجاوز تأخير الاستجابة مهلة زمنية "Response Time Out" يعود الجهاز الرئيسي "Master" بإعادة الإرسال لنفس الطلب و يكون الناقل "Physical Line" مشغولاً خلال عملية ارسال الطلب "REQUEST" كما هو موضح بالشكل (4).

3-5 أنماط بروتوكول MODBUS التسلسلي:

يوجد نمطين مختلفين لبروتوكول MODBUS التسلسلي هما : (RTU) و (ASCII) وهي تحدد محتويات بنات حقول الرسالة المرسله بشكل تسلسلي على الناقل، و يُحدد كيفية تعبئة المعلومات في حقول الرسالة وفك تشفيرها.

يجب أن يكون وضع الإرسال هو نفسه لجميع الأجهزة على ناقل MODBUS التسلسلي، و يجب على المستخدمين إعداد الأجهزة على وضع الإرسال المرغوب (RTU) أو (ASCII)، علماً أنّ الوضع الافتراضي للأجهزة هو (RTU).

1-3-5 نمط الإرسال (RTU):

عندما تتواصل الأجهزة على ناقل MODBUS التسلسلي باستخدام وضع (RTU)، فإن كل بايت (8) بت في الرسالة يحتوي على حرفين سداسي عشريين (hex) بطول (4) بت، و تسمح هذه الميزة أن تكون كثافة الأحرف أكبر ممّا يعني إخراج بيانات أفضل لنفس معدل النقل مقارنةً مع النمط (ASCII)، و يجب إرسال كل رسالة بسلسلة من الأحرف.[2]

- التسبيق (11) بت لكل بايت في وضع (RTU): بت التكافؤ (parity) بطول (1) بت، و يكون وضع التكافؤ الافتراضي (even)، و في حال عدم وضع بت تكافؤ سنقوم بوضع (2 bits) للتوقف، و يتم نقل الأحرف بالتسلسل، حيث يُرسل كل حرف أو بايت بهذا الترتيب من اليسار إلى اليمين، أي من الأقل أهمية (LSB) إل الأكثر أهمية (MSB) كما يظهر في الشكل (5).

Start	1	2	3	4	5	6	7	8	Par	Stop
-------	---	---	---	---	---	---	---	---	-----	------

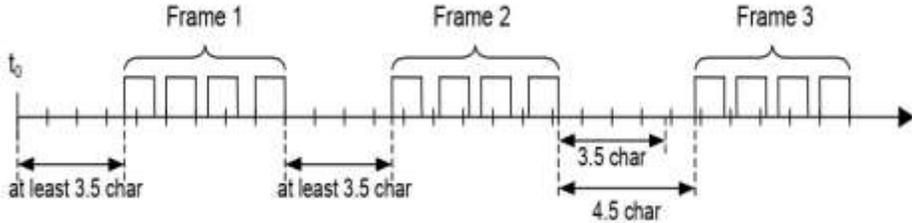
الشكل (5): تسلسل البتات لبروتوكول MODBUS-RTU.

- إطار MODBUS-RTU: يُظهر الشكل (6) حقول إطار MODBUS-RTU ، بحجم (256 Byte) كحد أقصى.

Slave Address	Function Code	Data	CRC
1 byte	1 byte	0 up to 252 byte(s)	2 bytes CRC Low CRC Hi

الشكل (6): إطار MODBUS RTU.

- **تأطير رسائل MODBUS RTU:** يتم وضع رسالة MODBUS بواسطة جهاز الإرسال في إطار له نقطة بداية ونهاية معروفة، و يسمح هذا للأجهزة التي تتلقى إطاراً جديداً معرفة وقت اكتمال الرسالة، وفي وضع (RTU) يتم فصل إطارات الرسائل بفواصل زمني لا يقل عن (3.5 character) كما في الشكل (7)، و يتوضع الفاصل الزمني في بداية و نهاية الإطار.



الشكل (7): آلية ارسال الأطر في MODBUS-RTU.

ملاحظة: بالنسبة لمعدلات الإرسال التي تزيد عن (19200) بت في الثانية يُوصى باستخدام قيمة (750) ميكرو ثانية للمهلة بين الأحرف (1.5 char) وقيمة (1.750) ميلي ثانية للفواصل الزمن (3.5 char). [2].

6- ناقل (RS485) لبروتوكول MODBUS التسلسلي:

على المستوى الفيزيائي يستخدم MODBUS التسلسلي نواقل مختلفة مثل (RS485، RS232)، و يعتبر الناقل (RS485) ثنائي الأسلاك الأكثر شيوعاً، و سنذكر فيما يلي أهم متطلبات ناقل "RS485" [2]:

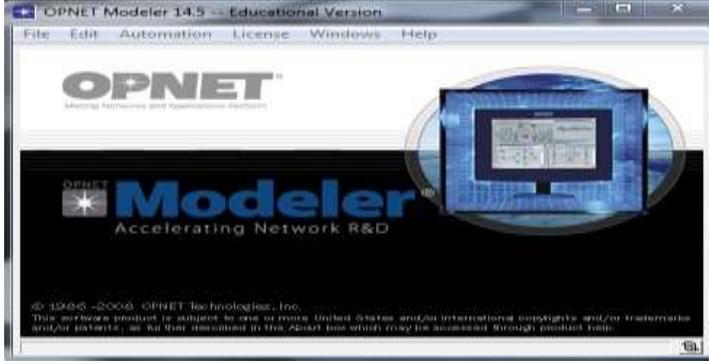
- a) **عدد الأجهزة:** بالنسبة لأي نظام متعدد النقاط (EIA / TIA-485) ، سواء في تكوين 2 سلك أو 4 أسلاك ، يكون عدد الأجهزة التي يمكن وصلها بدون مكرّر (Repeater) هو (32) جهاز.

- (b) الطوبولوجيا: يحتوي تكوين RS485-MODBUS بدون مكرر على كابل (trunk) واحد، يتم على طوله توصيل الأجهزة بشكل مباشرة (daisy chaining) أو بواسطة كابلات اشتقاق قصيرة (derivation cables)، و يجب توصيل طرفي الناقل الرئيسي عند انتهاء الخط بمقاومات.
- (c) طول الناقل: يجب أن يكون طول الناقل (trunk) محدوداً، و يعتمد الحد الأقصى للطول على معدل الارسال بالبت، فالحد الأقصى لمعدل ارسال [9600 bps] مثلاً هو (1) كيلومتر و طول كابلات الاشتقاق قصير نسبياً و لا يتجاوز (20) متر.
- (d) التأريض: يجب توصيل السلك المشترك "Common" مباشرة بالأرض ويفضل عند نقطة واحدة فقط للناقل بأكمله، و بشكل عام تكون هذه النقطة عند الجهاز الرئيسي "Master".
- (e) إنهاء الناقل: الانعكاس في الناقل هو نتيجة لانقطاع الممانعة الذي تعانيه الموجة المتقلة أثناء انتشارها في الناقل، و لتقليل الانعكاسات من نهاية ناقل (RS485) يلزم وضع نهاية خط (LT)، و يتم وضع مقاومتي إنهاء عند طرفي الناقل تصل بين السلكين (D0) و (D1)، و تكون قيمة المقاومة (150 ohm) و باستطاعة (0.25 watt)، و يتم وصل مكثف قطبي (1) نانو فاراد (10) فولط.

7- محاكي الشبكات OPNET:

إنّ OPNET عبارة عن أداة من شركة MIL3 طوّرها الطالب Alain Cohen في عام 1986 وهو اختصار لـ Optimized Network Engineering Tools أي أدوات هندسة الشبكات المحسّنة، وهو أحد أشهر المحاكيات الشبكيّة وأكثرها شعبية بسبب استخدامه الكبير والواسع في مجالي الصناعة والأبحاث الشبكية، وهو نظام هندسي قادر على محاكاة شبكات الاتصال الضخمة مع نمذجة تفصيليّة للبروتوكولات والتطبيقات

والأجهزة وتحليل الأداء، و تمّ استخدام نسخة OPNET الأكاديمية ذات الاصدار (14.5) في بحثنا هذا و تظهر واجهة البرنامج بالشكل(8).[3].



الشكل (8): واجهة برنامج OPNET اصدار (14.5) المُستخدم.

7-1 أهم خصائص برنامج OPNET:

(a) دورة النمذجة والمحاكاة: إنّ OPNET يُوفر أداة فعالة لمساعدة المستخدم في تحقيق المراحل الثلاث الأولى من طور التصميم وهي: بناء نموذج، تنفيذ المحاكاة، تحليل الخرج.

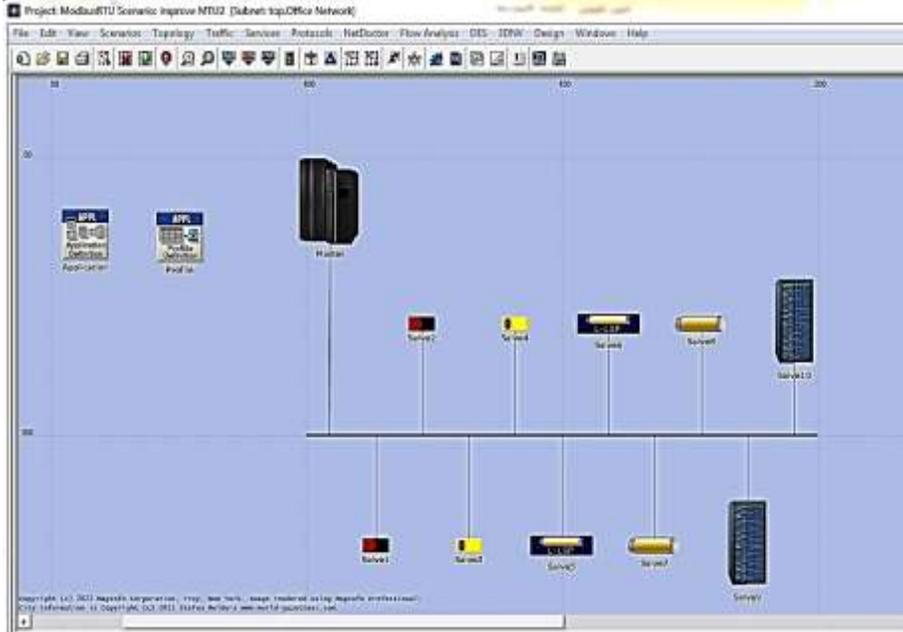
(b) النمذجة الهرمية: إنّ OPNET يوظف بنية هرمية من أجل النمذجة وكل مستوى من الهرمية يصف مفاهيم مختلفة من النموذج الكامل الذي يتم مُحاكاته، حيث أنّ النماذج في OPNET تُبنى بشكل هرمي، النماذج يمكن أن تُبنى إما من الأعلى للأسفل أو من الأسفل للأعلى وكلّ مستوى يمثّل التركيب الداخلي ووظيفة المستوى الأعلى، و هي (مستوى الشبكة و مستوى العقدة و مستوى العملية).[3].

(c) تُخصّص لشبكات الاتصال: OPNET يملك مكتبات مُفصّلة تزود دعم كبير لبروتوكولات الاتصال و يتيح للباحثين بتعديل النماذج الموجودة أو تطوير نماذج جديدة خاصة بهم.

(d) توليد المحاكاة بشكل أوتوماتيكي: إنّ نموذج OPNET يترجم إلى شيفرة مصدريّة قابلة للتنفيذ وهذه الشيفرة الخاصة بمحاكاة الأحداث المنقطّعة، يتم تنفيذها ببساطة والحصول على نتائج الخرج.

8- نمذجة مكونات شبكة صناعية لبروتوكول MODBUS التسلسلي:

نقوم بإنشاء شبكة ذات طوبولوجيا ناقل تسلسلي مُكوّنة من (11) عقدة قابلة للاختبار، واحدة منها هي "Master" و الأخرى من النوع "Slaves" و تمّ وضع عدة أنواع من العقد التابعة مثل (الحساسات الرقمية والتمائثية و المشغلات الرقمية والتمائثية و أجهزة التحكم المنطقية PLC التابعة) تتبادل البيانات وفق نمط الارسال MODBUS-RTU كما يظهر في الشكل (9).

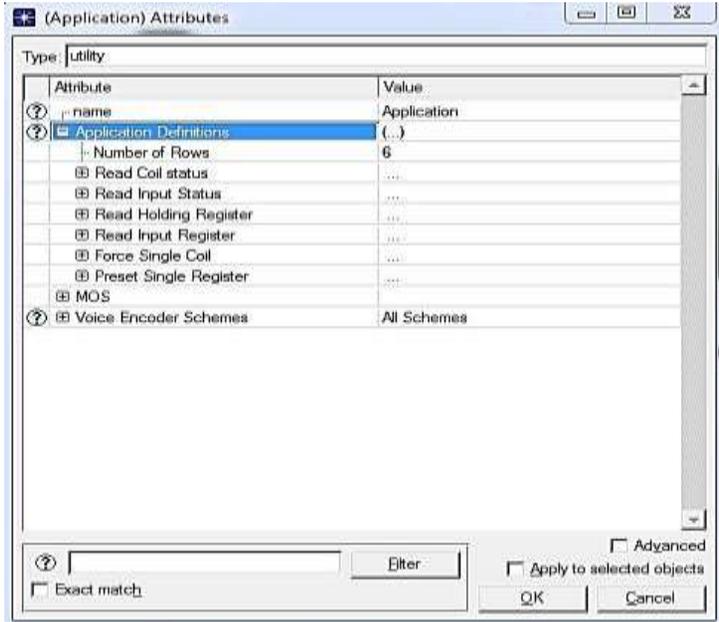


الشكل (9): شبكة بروتوكول MODBUS التسلسلي باستخدام برنامج OPNET.

ملاحظة هامة جداً: لا تحتوي مكتبة OPNET ناقل بروتوكول MODBUS بمختلف أنواع العقد المتصلة به من حساسات و مشغلات و أجهزة تحكم منطقيّة PLC و ناقل RS485 لنقل أطر البيانات ذات النمط RTU المُستخدم، لذلك قمنا من خلال هذا

البحث بنمذجة هذه المكونات من خلال عملية تعديل خصائص العقد المُختارة لتناسب مع خصائص شبكة ناقل MODBUS التسلسلي، مُقدمين نموذج لشبكة MODBUS قابلة لعمليات التحليل و الاختيار.

- فيما يلي سوف نوضح خصائص كل عقدة مستخدمة في هذا النموذج:
خصائص الجهاز (Application): هو جهاز يتيح نمذجة التطبيقات وفق الشبكة المدروسة و بروتوكول الاتصال MODBUS المُستخدم، وتم نمذجة ستة تطبيقات فقط علماً أنه يوجد العديد من التطبيقات التي يدعمها بروتوكول MODBUS و تختلف عن بعضها بحجم و عنوان و وظيفة الطلب المرسل من العقدة الرئيسية إلى إحدى العقد التابعة كما يظهر في الشكل(10):



الشكل (10) : تعريف التطبيقات الستة المستخدمة في شبكة MODBUS.

حيث أن:

1. التطبيق (Read Coil Status): يدل على إرسال طلب من العقدة الرئيسية

للعقدة التابعة لقراءة حالة خرج رقمي و يُرمز للوظيفة بالعنوان (01) بشكل

ستة عشري بطول واحد بايت.

2. التطبيق (Read Input Status): يدل على إرسال طلب من العقدة الرئيسية للعقدة التابعة لقراءة حالة دخل رقمي و يُرمز للوظيفة بالعنوان (02) بشكل ستة عشري بطول واحد بايت.
3. التطبيق (Read Holding Register): يدل على إرسال طلب من العقدة الرئيسية للعقدة التابعة لقراءة حالة خرج تماثلي و يُرمز للوظيفة بالعنوان (03) بشكل ستة عشري بطول واحد بايت.
4. التطبيق (Read Input Register): يدل على إرسال طلب من العقدة الرئيسية للعقدة التابعة لقراءة حالة دخل تماثلي و يُرمز للوظيفة بالعنوان (04) بشكل ستة عشري بطول واحد بايت.
5. التطبيق (Force Single Coil): يدل على إرسال طلب من العقدة الرئيسية للعقدة التابعة لكتابة أمر لخرج رقمي و يُرمز للوظيفة بالعنوان (05) بشكل ستة عشري بطول واحد بايت.
6. التطبيق (Preset Single Register): يدل على إرسال طلب من العقدة الرئيسية للعقدة التابعة لكتابة أمر لخرج تماثلي و يُرمز للوظيفة بالعنوان (06) بشكل ستة عشري بطول واحد بايت.

تمّ تعيين كل تطبيق وفق الخصائص الموضّحة بالشكل(11):

Attribute	Value
Command Mix (Get/Total)	100%
Inter-Request Time (seconds)	constant (0.00175)
File Size (bytes)	constant (16)
Symbolic Server Name	FTP Server
Type of Service	Best Effort (0)
RSVP Parameters	(...)
Back-End Custom Application	Not Used

الشكل (11): ضبط خواص التطبيق المُعرف في برنامج OPNET.

❖ الخاصية "Inter-Request Time" تدل على الفترة الزمنية التي تفصل بين الأطر المرسله و تم تحديدها بالقيمة (1.750) ميلي ثانية لتتناسب خصائص بروتوكول MODBUS.

❖ الخاصية "File Size" تدل على حجم الإطار المرسل مقدراً بالبايت و يُمكن تحديد حجمه ضمن المجال (8 Byte) إلى القيمة (256 Byte) آخذين بعين الاعتبار حجم ترويسة إطار MODBUS-RTU و التي تُقدر بـ (4 [Byte]).

خصائص الجهاز (Profile): يتم بالبداية تعيين ستة ملفات تعريف حسب الرمز المُعتمد لترميز الوظائف في شبكات MODBUS حيث يحصل كل تطبيق على رمز مُؤلف من خانيتين بشكل ستة عشري، ويتم ضبط ملف التعريف الأول (01) على التطبيق (Read Coil Status) و يتم تحديد السمات كالتالي:

❖ السمة (Start Time Offset) على القيمة (Constant=1[s]): هذه السمة تُحدد الفترة الزمنية بين نهاية أحد التطبيقات وبداية التطبيق التالي.

❖ السمة (Duration) على القيمة (End of the Profile): تُعبّر عن الحد الأقصى من الزمن المسموح به لجلسة التطبيق قبل أن يتم إحباطها.

❖ السمة (Start Time) على القيمة (Constant=10[s]): تُحدد زمن التأخير لبداية المحاكاة.

ملاحظة: بالاعتماد على السمة الأولى و الثالثة سوف تحتاج كل عقدة تعمل على مثل هذا النوع من التطبيقات لزمن تأخير يُقدر بـ (11) ثانية. يتم تحديد ملفات تعريف باقي التطبيقات بنفس الطريقة كما في الشكل (12):

Profile Name	Applications	Operation Mode	Start Time (seconds)	Duration (seconds)	Repeatability
01 01	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time
02 02	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time
03 03	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time
04 04	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time
05 05	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time
06 06	(.)	Serial (Ordered)	constant (10)	End of Simulation	Once at Start Time

الشكل (12): ملفات تعريف تطبيقات شبكة MODBUS الستة.

خصائص الجهاز الرئيسي (Master): يمثل هذا الجهاز العقدة الرئيسية في الشبكة و التي تدير عمليات الارسال و الاستقبال في شبكات MODBUS و تم ضبط خصائصها كالتالي:

❖ السمة (wkatn): يتيح هذا النوع من الأجهزة تعريف تطبيقات مختلفة تدعمها العقدة الرئيسية، حيث لم يتم بناء هذه العقدة وفق مستويات برنامج OPENT الهرمية، وإنما تم تعديل بعض الخصائص بما يلائم مزايا العقد المستخدمة في الشبكة المدروسة، و إضافة بعض السمات على مستوى العقدة مثل السمة "Type" التي تحدد نوع الجهاز المستخدم و تم إضافة السمة "Device" لتحديد نوع الوظيفة التي تدعمها العقدة.

❖ السمة (Application Supported Profile): تُحدد أسماء جميع ملفات التعريف التي تم تمكينها على هذه العقدة لدعم جميع أنواع التطبيقات المتاحة في نموذج MODBUS.

❖ السمة (Application Supported Services): تتيح هذه السمة للعقدة استقبال جميع أنواع التطبيقات المتاحة من العقدة الأخرى و قد تم ضبطها على القيمة (ALL).

خصائص العقدتين (Salve1+Salve2): تم تعيين هاتين العقدتين كحساسين رقميين و بتالي سيتم ضبط السمات كالآتي:

- ❖ السمة (Application Supported Profile): على ملف التعريف (02) الذي يعبر عن تطبيق (Read input status).
- ❖ السمة (Application Supported Services): على القيمة ALL.
- ❖ السمة (Type Device): على القيمة (digital sensor).

خصائص العقدتين (Salve3+Salve4): تم تعيين هاتين العقدتين كحساسين تماثليين و بتالي سيتم ضبط السمات كالتالي:

- ❖ السمة (Application Supported Profile): على ملف التعريف (04) الذي يعبر عن تطبيق (Read input register).
- ❖ السمة (ApplicationSupported Services): على القيمة ALL.
- ❖ السمة (Type Device): على القيمة (analog sensor).

خصائص العقدتين (Salve5+Salve6): تم تعيين هاتين العقدتين كمشغلين رقميين و بتالي سيتم ضبط السمات كالتالي:

- ❖ السمة (Application Supported Profile): سيتم ضبطها على ملفي التعريف (01) و (05) أي (Read Coil Status) و (Force Single) (Coil) بالترتيب.
- ❖ السمة (Application Supported Services): على القيمة ALL.
- ❖ السمة (Type Device): على القيمة (Digital Actuator).

خصائص العقدتين (Salve7+Salve8): تمّ تعيين هاتين العقدتين كمشغلين تماثلين و بتالي سيتم ضبط السمات كتالي:

- ❖ السمة (Application Supported Profile): سيتم ضبطها على ملفي التعريف (03) و (06) أي (Read Holding Registers) و (Preset Single Registers) بالترتيب.
- ❖ السمة (ApplicationSupported Services): على القيمة ALL.
- ❖ السمة (Type Device): على القيمة (Analog Actuator).

خصائص العقدتين (Salve9+Salve10): تمّ تعيين هاتين العقدتين كمتحكمين منطقيين قابلين للبرمجة (PLC) و بتالي سيتم ضبط السمات كتالي:

- ❖ السمة (Application: Supported Profile): سيتم ضبطها على ملفات التعريف الستة.
- ❖ السمة (Application Supported Services): على القيمة ALL تتيح هذه القيمة استقبال كافة طلبات التطبيقات المستخدمة في شبكة MODBUS-RTU المدروسة.

خصائص الناقل (RS485): هو الناقل الرئيسي في شبكة MODBUS-RTU، و تمّ تحديد نوعه "coax-adv" و هذا النوع يُتيح عدة سمات نستطيع من خلالها رصد العديد من البارامترات الهامة مثل (حالة حصول تصادم بين الأطر و كذلك يمكن معرفة فيما إذا تم قبول الإطار و يمكن حساب عدد الأخطاء في بتات الإطار و يمكن أيضاً مراقبة زمن تأخير الارسال و الانتشار للإطار عبر الناقل) و غيرها من البارامترات الهامة، و لقد تمّ ترك السمات على القيم الافتراضية باستثناء البعض منها و نذكر فيما يلي السمات التي تم ضبطها ليُمثل هذا الناقل خصائص ناقل RS485 المُستخدم في شبكات MODBUS- RTU الصناعية.

- ❖ السمة(ber): تمّ ضبطها على القيمة (10^{-6}).

❖ السمة (data rate): تم ضبطها على القيمة (19200 bps).

❖ السمة (delay): تم ضبطها على القيمة (0.001) ثانية.

❖ السمة (packet format): تم ضبطها على القيمة (Modbus-RTU).

❖ السمة (thickness): تم ضبطها على القيمة (2) ملم.

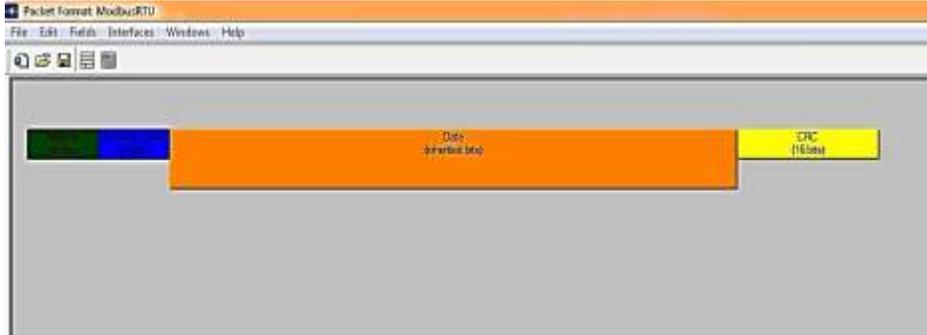
إطار MODBUS-RTU: يُتيح برنامج OPNET بناء إطار بروتوكول اتصال معين، و تم تشكيل إطار MODBUS-RTU وفق الخطوات الآتية:

❖ من قائمة "File" نختار "New" و من ثم نختار محرر الأطر " Packet "

"Format"، من النافذة نختار "Create New Field".

❖ نقوم بإنشاء إطار MODBUS-RTU المؤلف من أربعة حقول كما في الشكل

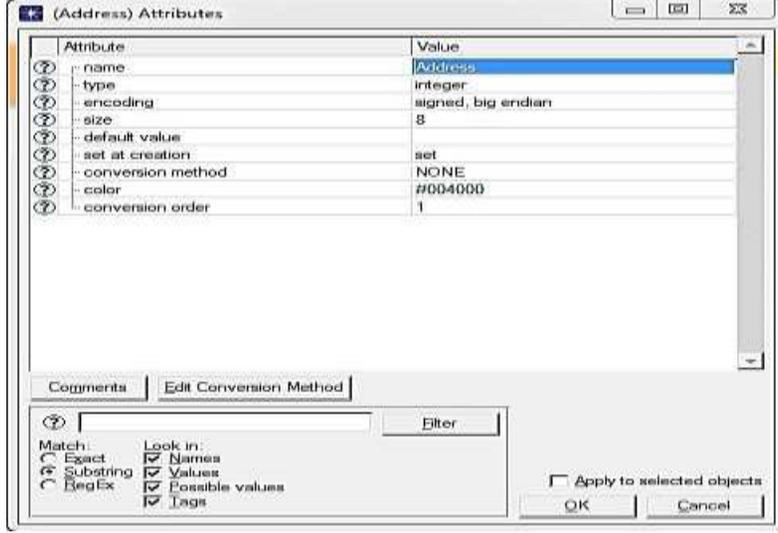
(13).



الشكل (13): حقول إطار MODBUS-RTU الأربعة في برنامج OPNET.

❖ نقوم بتحرير أربعة حقول ومن ثم نحدد خصائص كل حقل "attribute"

كما في الشكل (14) و الذي يُظهر خصائص الحقل "Address".



الشكل (14): خصائص حقل "Address" في إطار MODBUS-RTU.

ملاحظة : تتشابه كل من الحقول "Address" و "Function Code" و "CRC" بنوع الخاصية فجميعها من النوع "integer" و تختلف فقط بالحجم حسب خصائص بروتوكول MODBUS-RTU.

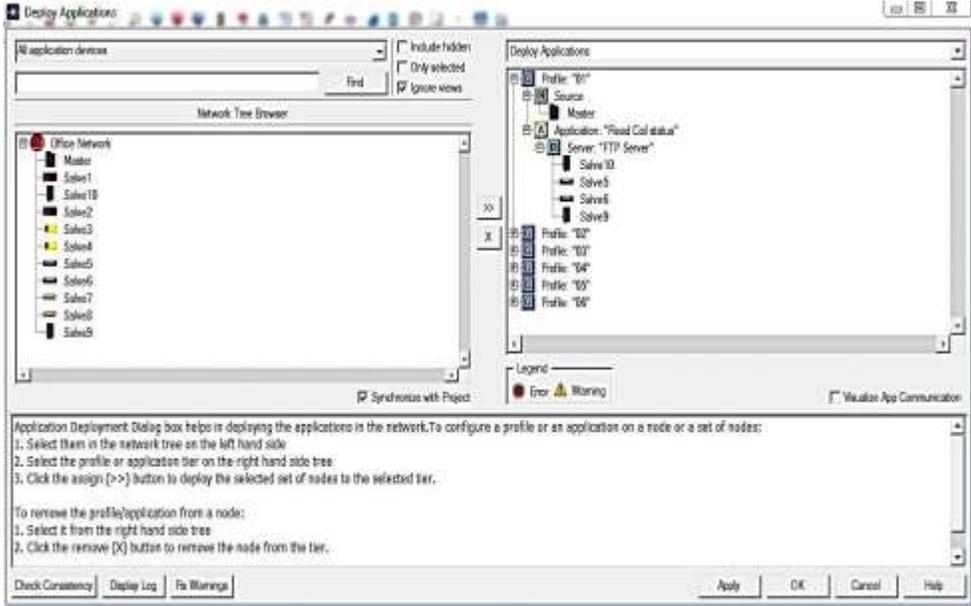
❖ حقل "Data" يتم ضبط خاصية "Type" على النوع "Packet" ومن ثم اختيار "Inherited"، حيث يتيح خيار "Inherited" تعيين حجم الحقل على حجم البيانات الفعلي وفق التطبيق المستخدم، و يتم ضبط الترميز "encoding" على النوع "signed.big endian" الذي يحافظ على تسلسل البيانات المرسل بجهة الاستقبال.

❖ نقوم بحفظ الإطار باسم "ModbusRTU" و من ثم نعود إلى خصائص ناقل RS485 ونقوم بضبط الخاصية "Packet Formats" على الوضع "ModbusRTU" حيث نقوم بجعلها "supported".

ملاحظة هامة: قبل تشغيل الشبكة المُصمّمة لا بدّ من جعل العقدة الرئيسية هي التي تقوم بإرسال جميع أوامر التطبيقات للعقد التابعة الأخرى وفق نمط الإرسال

(Master\Salve) المُعتمد في بروتوكول MODBUS-RTU و تم ضبط ذلك بالخطوات التالية:

❖ من تبويبه (Protocols) نختار (Applications) و من ثم (Deploy (Defined Application) و نقوم بضبط خواص العقد كما في الشكل (15).



الشكل (15): ضبط خواص عقد شبكة MODBUS لتحديد "Master" و "Salves".

9- بارامترات أداء شبكة MODBUS التسلسلي:

يرتبط مفهوم أداء الشبكات الصناعية بعدد كبير من البارامترات نذكر منها (الاستخدامية للشبكة "Utilization"، و مقدار الخطأ في البت "BER"، و نسبة عدد البتات المرسلة إلى المستلمة، و زمن تأخير الأطر في الشبكة "End To End Delay"، و عدد التصادمات للأطر "Collisions"، و زمن الاستجابة "Time Response"، و الانتاجية "Throughput") و غيرها.

يتيح برنامج OPNET دراسة الكثير من الإحصائيات المرتبطة بالأداء حيث تُقسم إلى ثلاثة أنواع رئيسية:

1. (Global Statistics): تتيح دراسة بارامترات تؤثر على كامل الشبكة.

2. (Node Statistics): يمكن من خلالها دراسة بارامترات تؤثر في العقدة المرتبطة بالشبكة.

3. (Link Statistics): يمكن من خلالها دراسة بارامترات تؤثر في خطوط النقل التي تربط بين عقد الشبكة.

ملاحظة: أثناء اختيار بارامترات الأداء المرتبطة بالشبكة المدروسة لا بدّ من توخي الحذر لأنّ البرنامج يتيح حرية اختيار الاحصائيات من قبل المُستخدم بشكل عام لذلك قد لا تظهر أية نتائج مرتبطة بالإحصائيات المختارة نتيجة عدم ارتباطها بخصائص الشبكة المدروسة، لذا لا بد من اختيار الاحصائيات المرتبطة بأداء الشبكة بشكل دقيق.

العوامل المؤثرة في بارامترات الأداء: يوجد العديد من العوامل المؤثرة في بارامترات أداء الشبكات الصناعية ذات الناقل MODBUS تمّ دراستها في العديد من الأبحاث المرجعية نذكر منها (طوبولوجيا الشبكة ، عدد العقد في الشبكة ، طول و نوعية الناقل) و غيرها من العوامل الأخرى، و تمّ اختيار العامل (MTU) في دراستنا و الذي هو اختصار للعبارة (Maximum Transmission Unit) و يُعرّف بوحدة الإرسال العظمى أي حجم أكبر بيانات مرسل في حقل الـ (Data) للإطار و هو مُرتبط بالقيمة الأعظمية لحجم الإطار المرسل وفق بروتوكول MODBUS-RTU، و يرتبط العامل (MTU) بمجموعة عوامل أخرى بالعلاقة الرياضيّة الآتية:

$$T_{transmission} = \frac{D (H + MTU)}{MTU * R}$$

حيث أنّ:

- ($T_{transmission}$): زمن الإرسال اللازم لإرسال المعلومات بشكل كامل من المنبع.
- (D): عدد البيانات الكلي أو حجم (Data).
- (H): حجم الترويسة للبروتوكول المستخدم.
- (R): معدّل الإرسال.

نلاحظ من العلاقة الرياضية السابقة وجود العامل (MTU) في البسط و المقام و بتالي لا بد من إجراء اختبار دقيق لتحديد قيمة مناسبة له، فالقيمة الصغيرة نسبياً تعكس سلباً على الأداء نتيجة زيادة حجم الترويسات للأطر المرسله مما يؤدي إلى انخفاضه، و كذلك زيادة حجمه نسبياً سيزيد من طول الإطار و بتالي سيؤثر على بارامترات أخرى مرتبطة بالأداء و بتالي انخفاضه أيضاً، لذا لا بدّ من اختيار قيمة وسط للمقدار (MTU) تحقّق أفضل أداء للشبكة.

البارومتريات المرتبطة بأداء ناقل MODBUS الصناعي نذكر منها:

1. (Response Time): و هو من النوع (Global Statistics) و يُمثل الزمن

اللازم لإرسال الطلب واستلام الاستجابة، حيث يتم قياس زمن التأخير بين لحظة ارسال العقدة الرئيسية (Master) الطلب إلى العقدة التابعة (Slave1) ولحظة تلقي الاستجابة و يُقاس هذا التأخير الزمني بالثانية و يتناسب عكساً مع أداء الشبكة فبنقصانه يزداد الأداء.

2. (End To End Delay): و هو من النوع (Global Statistics) و يُمثل

الزمن التأخير (end to end) الفاصل بين لحظة ارسال الرسالة من المنبع إلى لحظة استقبالها من قبل الوجهة، و من أجل الحصول على أداء أفضل لا بد من تخفيضه قدر الإمكان.

3. (Utilization): و هي من النوع (Link Statistics) و تمثل النسبة المئوية

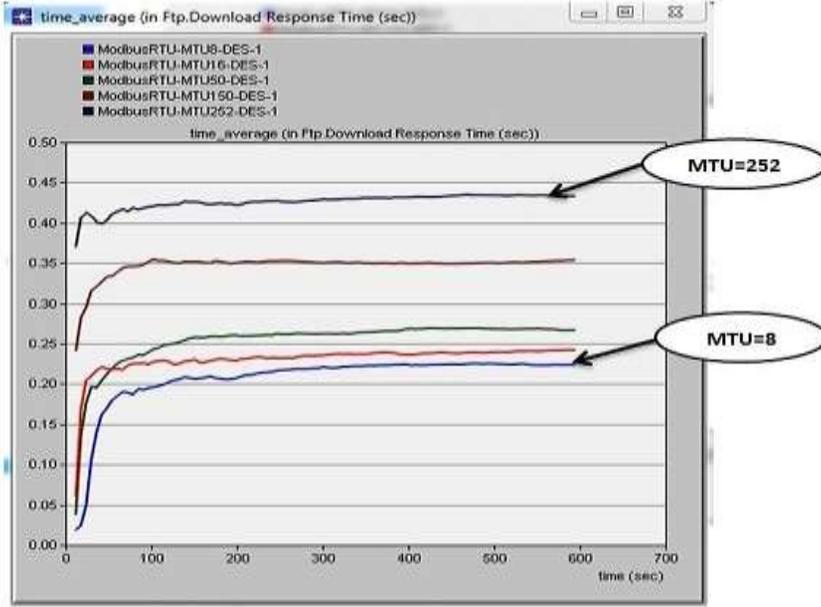
المُستهلة لعرض النطاق الترددي للنقل الرئيسي و بازياد النسبة يزداد الأداء.

10- دراسة و تحليل النتائج المرتبطة ببارامترات شبكة MODBUS التسلسلي:

تمّ تشغيل عملية المحاكاة على خمسة سيناريوهات مختلفة ، تمّ فيها زيادة قيمة العامل MTU بالترتيب (252-150-50-16-8) بايت، و تمّ تحديد زمن المحاكاة بمقدار (10) دقائق.

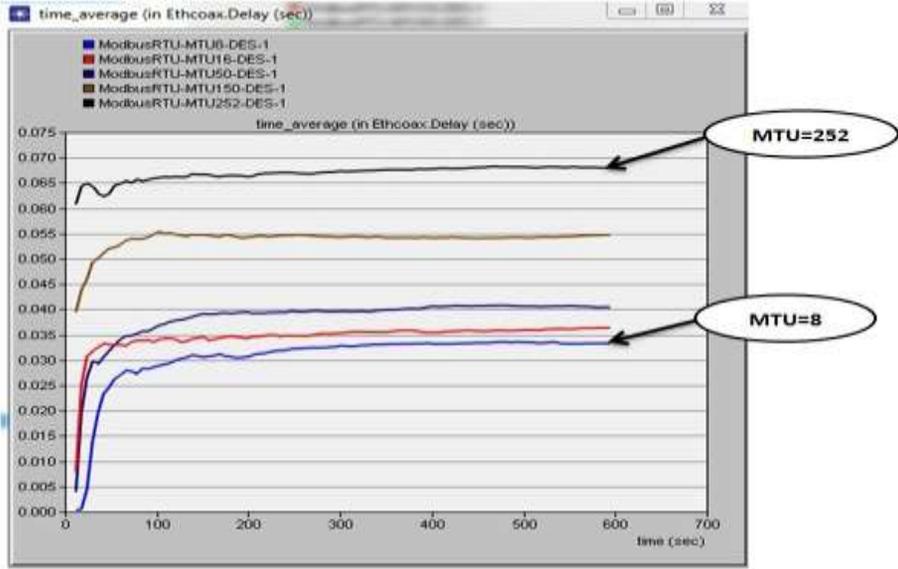
-البارامتر (Response Time): نلاحظ أنّه مع ازدياد حجم (MTU) يزداد زمن

الاستجابة فعند القيمة (MTU=8 [Byte]) سجل قيمة (0.2) ثانية و ازدادت بشكل طفيف مع ازدياد الحجم لتصل للقيمة (0.45) ثانية عند (MTU=252 [Byte])، كما هو مُوضح بالشكل(16).



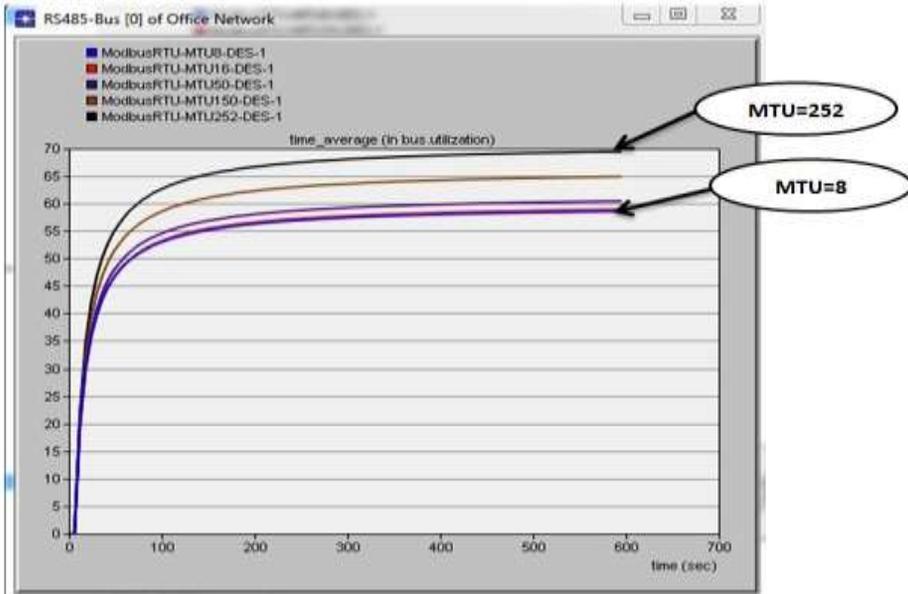
الشكل (16): مقارنة قيم البارامتر "Response Time" عند احجام MTU مختلفة.

- البارامتر (End To End Delay): بشكل مشابه لبارامتر زمن الاستجابة سجل تزايداً طفيفاً بدأ من القيمة (30) ميلي ثانية عند (MTU=8 [Byte]) و وصل للقيمة (70) ميلي ثانية عند (MTU=252 [Byte])، كما يظهر في الشكل(17).



الشكل (17): مقارنة قيم البارامتر "End To End Delay" عند احجام MTU مختلفة.

- البارامتر (**Utilization**): شهد هذا البارامتر ازدياد مع ازدياد حجم (MTU) فسجل قيمة (55%) عند أول سيناريو و قيمة (70%) عند خامس سيناريو كما يظهر في الشكل (18).



الشكل (18): مقارنة قيم البارامتر "Utilization" عند احجام MTU مختلفة.

11- النتائج و الاستنتاجات:

على الرغم من ازدياد أزمنة التأخير بازدياد حجم الإطار المُرسَل نتيجة زيادة العامل MTU إلا أنه يبقى في مستويات مقبولة بالنسبة لمتطلبات الزمن الحقيقي في الشبكات الصناعية و الذي يُحدد قيمة عظمى لتأخير زمن الاستجابة بمقدار (0.5) ثانية، من جهة أخرى لاحظنا ارتفاع في استخدامية الناقل بقدر (15%) و بتالي رفع أداء الشبكة الصناعية لبروتوكول MODBUS التسلسلي.

إنّ استخدام حساسات و مشغلات صناعية ذات دقة عالية نسبياً و بحجم معلومات من رتبة (200 [Byte]) لن يخل بمُتطلبات الزمن الحقيقي بل له أثر إيجابي في زيادة استخدامية الناقل و بتالي رفع أداء الشبكات الصناعية MODBUS ، و على الرغم من عدم تأثر الناقل بأقصى حجم للإطار عند (MTU = 252 [Byte]) إلا أنه لا يُنصح بالوصول لهذه القيمة نتيجة الاقتراب الكبير من القيمة الحديّة التي تخل بشروط العمل في الزمن الحقيقي للشبكات الصناعية فتبقى نماذج المحاكاة مثاليّة من ناحية التصميم و لا يظهر فيها عيوب الوصلات و الاسلاك التي تُسبب تأخير لا بدّ منه في الشبكات الصناعية الواقعية.

مما سبق يُنصح بضبط أحجام الأطر للعقد المتصلة بالشبكة ضمن المجال من (150) إلى (200) بايت بحيث تُسبب زيادة في استخدامية الناقل و بتالي الأداء و بشكل لا يزيد من أزمنة التأخير لتبقى في مستويات مقبولة لعمل الشبكات الصناعية في الزمن الحقيقي.

12- التوصيات و الآفاق المستقبلية:

- ✓ يمكن استخدام نموذج الشبكة المصمّم لدراسة عوامل أخرى مؤثرة في بارامترات الأداء.
- ✓ إجراء نماذج لشبكات MODBUS/TCP و تحليل أدائها.
- ✓ نمذجة شبكات صناعية واقعية باستخدام برنامج OPNET و اختبار أدائها.
- ✓ تصميم منظومة عملية للمقارنة بين أنماط ارسال MODBUS و اختبار أدائها.

المراجع:

- [1] KUMER SEN, S2014–Fieldbus and Networking in Process Automation. CRC Press, New York ,439p.
- [2] MODBUS.Org,2006–MODBUS over Serial Line Specification and Implementation Guide. US,44p.
- [3] LU YANG,H2012–Unloking The Power of OPNET Modeler. CAMBRIDGE, UK, 253p.
- [4] Yunyuan.Y, Meng.C,2020–An Improved Algorithm for Adaptive Communication Frame Length Based on Modbus Protocol. IEEE, Shenyang Institute of Technology, China,8p.
- [5] Gamess.E, Smith.B, and Francia.G,2020 – PERFORMANCE EVALUATION OF MODBUS TCP IN NORMAL OPERATION AND UNDER A DISTRIBUTED DENIAL OF SERVICE ATTACK. IJCNC, Florida,US, Vol.12, No.2,21p.
- [6] Mnaouer.A, Fujii.Y, Sekiguch.T,2017– Colored Petri Nets Based Evaluation of Transmission Procedures at a Fieldbus Data Link Layer Protocol.IEEE, Yokohama University,Brazil,12p.
- [7] Krupanek.B , Bogacz.R,2016– OPNET Modeler simulations of performance for multi nodeswireless systems. Silesian University of Technology, Poland,10p.
- [8] Kim.B, Lee.D, Choi.T,2015– Performance Evaluation for Modbus/TCP Using Network Simulator NS3.IEEE, Kyungpook National University, Korea,10p.
- [9] Künzel.G, Ribeiro.C, Pereira.C,2014– A Tool for Response Time and Schedulability Analysis in Modbus Serial Communications. IEEE, Federal Institute of Education, Brazil,12p.

- [10] Hao.J, Wu.J, Guo.C,2011– Modeling and Simulation of CAN Network Based on OPNET. IEEE, Naval University, China,10p.
- [11] Oh.E,2009– Study of Network Design Factors That Influence Industrial Fieldbus Network–Based System Integration . The Ohio State University, Korea,84p.
- [12] Ali.Q,2007– Measurements and Performance Analysis of Industrial Ethernet. IEEE, University of Mosul, Iraq,16p.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات المقادة باستخدام التحكم الشعاعي

م. رغبة الابراهيم* د. شفيق باصيل** د. بسام عطيه***

*طالبة دراسات عليا (دكتوراه)، في قسم هندسة الطاقة الكهربائية، كلية الهندسة الميكانيكية والكهربائية، جامعة

البعث

**أستاذ في قسم هندسة الطاقة الكهربائية، كلية الهندسة الميكانيكية والكهربائية، جامعة البعث

***أستاذ في قسم هندسة الطاقة الكهربائية، كلية الهندسة الميكانيكية والكهربائية، جامعة تشرين

المخلص

يركز هذا البحث على تطبيق التحكم الشعاعي بالمحركات التحريضية المستخدمة في الرافعة الجسرية. كما تضمن شرحاً للتزامن في النظم الكهروميكانيكية، وكيفية تحقيقه عند اختلاف نسب توزيع الحمل على المحركات التحريضية، باستخدام المخطط الصندوقي للرافعة الجسرية الذي قمنا بتصميمه بناءً على النموذج الرياضي المحاكى للمحرك التحريضي المتحكم به بطريقة التحكم بالفيض الموجه لكل محرك على حدى.

أخيراً أثبتت نتائج المحاكاة في بيئة Matlab\Simulink فعالية طريقة التحكم الشعاعي بالفيض مقارنة مع التحكم السلمي، على الرغم من اختلاف نسب توزيع الحمل بين المحركات، فهي تؤمن استجابة سريعة، وكفاءة أكبر في التغلب على الاضطراب الخارجي. كما ساهمت في انخفاض الحملات الصدمية وبالتالي الاهتزازات بجسم الرافعة أثناء الأعمال القاسية إلى الحدود الدنيا المسموح بها وحققت زمن استجابة قياسي.

الكلمات المفتاحية: تحكم بالفيض الموجه، رافعة جسرية، نظم كهروميكانيكية، محرك تحريضي،

التزامن.

Synchronization Study of independent induction motors in multi- masses, multi-motors electromechanical Systems, driven by Using Vector Control

Abstract

This research focus on applying vector control to induction motors what is using in granty crane. It also includes explanation synchronization in the electromechanical systems, and How to achieve it when the load distribution ratios on the induction motors differ, using the block diagram of the gantry crane that we designed based on the simulated mathematical model of the induction motor controlled by field-oriented control way.

Finally, The simulation results in Matlab\Simulink environment proved the effectiveness of the proposed methodology compared to scalar control, despite the different load distribution ratios between the motors, as it provides a fast response, efficiency in overcoming external disturbance. It also contributed to reducing shock loads and thus vibrations in the crane body during harsh work to the minimum permissible limits and achieved a standard response time.

Key words: field-oriented control (FOC), gantry crane, electromechanical systems, induction motor, synchronization.

1- مقدمة

تُستخدم نظم القيادة الآلية متعددة المحركات في العديد من آليات الرفع والنقل والآليات الجبلية وتتصف هذه النظم باتصال مركب (متعدد الأبعاد) بين المحركات وعضو التشغيل من خلال وسائط نقل الحركة من جهة وعبر الهيكل المعدني للرافعة من جهة أخرى. بالإضافة إلى وجود عناصر المرونة من محاور وقنوات رفع الأحمال مع تأثير كل من عطالة المحركات وعطالة الكتل.

تتسبب هذه العناصر جميعها بحدوث حمولات وصددمات ديناميكية تفوق قيمها القيم المسموح بها للآلية الإنتاجية كما تؤدي إلى ظهور حالات عابرة اهتزازية تتسبب في تأخير العملية الإنتاجية وأحياناً في خروج الآلية عن العمل حيث أن استمرار هذه الحمولات مع اهتزازها يسرع من اهتراء الآلية الإنتاجية وخروجها من العمل قبل الفترة المخصصة لها.

وإذا تذكرنا أن الميزة الأساسية لعمل هذه الآليات هو النظام الدوري المتكرر والقصير الزمن بمعنى أنه لكل دورة عمل (إقلاع وفرملة وعكس اتجاه الحركة) ستظهر تلك الحمولات الصدمية التي تؤثر على القسمين الميكانيكي والكهربائي وخاصة أن غالبية هذه الآليات تستخدم المحركات التحريضية والتي تمثل عناصر غير خطية لذا فإن وجودها في نظام التحكم أيضاً يضيف مشكلة جديدة وهامة عند عمل هذه الآليات.

2- أهمية البحث

تبرز أهمية البحث في تصميم نموذج قيادة كهربائي بطريقة التحكم الشعاعي بالفيض الموجه لنظام كهروميكانيكي ثنائي الكتل - ثنائي المحركات التحريضية بهدف تطوير وتحسين جودة عمل هذه النظم وإنشاء نظم فعّالة وبسيطة وقليلة الاستهلاك للطاقة وتحقيق التوازن الفعلي عند عمل الروافع في الظروف القاسية المتمثلة بالتوزع غير المتساوي للحمولات على المحركات المستخدمة في الرافعة لأن أي اختلاف مهما كان صغيراً في سرعتها سينشأ عنه حمولات ديناميكية إضافية أخرى تسيء إلى عمل الآليات وإلى إنتاجيتها وتشكل خطورة على هيكلها المعدني.

3- طريقة البحث وخطواته

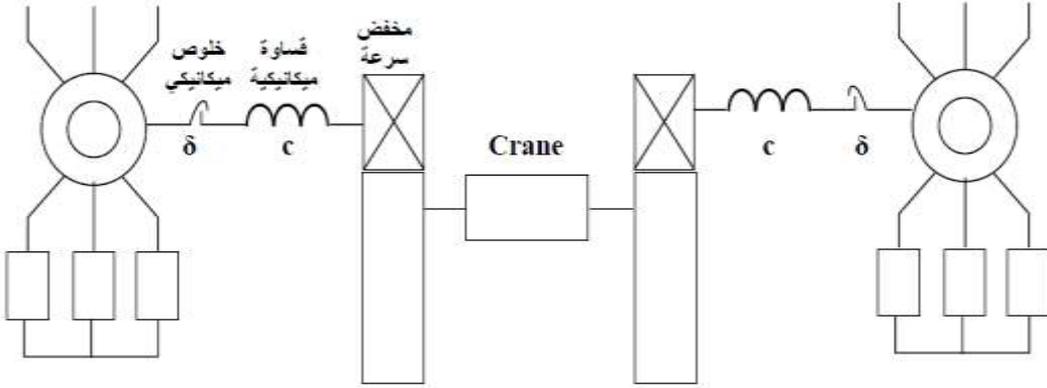
لتحقيق الهدف الأساسي أجرينا نمذجة للنظام الكهروميكانيكي مع المتحكمات الشعاعية بالمحركات التحريضية المستقلة المستخدمة، ومن ثم وظفنا مخطط المحاكاة هذا لقيادة الرافعة ثنائية القنال بالإضافة لدارات تغذية عكسية لتحقيق العمل المتزامن، ومن ثم درسنا توزيع الحمولة بشكل غير متساوي على القنالين وبنسب مختلفة لتكون الإشارات أقرب للواقع العملي، مما حسن أداء النظام وخفّض زمن الحالة العابرة بشكل كبير وهذا ما تؤكدته نتائج المحاكاة في بيئة .Matlab/Simulink

4- دراسة تزامن حركة عدة محركات في الرافعة

إن الصفة العامة لآليات الرفع هو عملها في نظام العمل الدوري المتكرر والقصير الزمن وبالتالي بالنسبة لها من المهم جداً التقليل من الحمولات الديناميكية وفي بعض الأحيان يتطلب العمل تحقيق الحد الأدنى لزمن الحالة العابرة وفي أحيان أخرى قد يتطلب الأمر تأمين الحركة التزامنية لعدة محركات مستقلة كما في الشكل (1) لأن أي اختلاف مهما كان صغيراً في سرعتها سينشأ عنه حمولات ديناميكية إضافية أخرى تسيء إلى عمل الآليات و إنتاجيتها وتشكل خطورة على هيكلها المعدني.

بشكل عام تتميز آليات الرفع بالعمل على سرعات منخفضة مع نظام عمل من النوع القاسي حيث أن زمن الاستمرارية بحدود /60%/ وبعده من المرات يتراوح بين /40-60/ مرة بالساعة إن ضرورة السرعات المنخفضة تأتي من ضرورة تنفيذ عمليات نقل ورفع الحمولات مع عمليات التنسيق بسلاسة وببطء ولكنها لا تتطلب تنظيمًا دقيقاً وناعماً بالإضافة لذلك يجب الانتباه بشكل جدي إلى الحمولات الديناميكية التي تنشأ عند العمل لذا فالمهم دوماً في هذه الآليات هو التقليل قدر الإمكان من زمن الحالة العابرة مع تأمين الإنتاجية الأعظمية ضمن مجالات التسارع

المطلوبة للآلية والتي لا يمكن تجاوزها كونها موضوعة أساساً وفق المواصفات العملية التكنولوجية [1].



الشكل (1) الدارة الحركية لآلية الرفع المرفئية (Crane)

تستخدم هذه الآلية بشكل واسع المحركات التحريضية المتميزة بالبساطة والموثوقية والرخص ولكن مشكلتها الأساسية هي صعوبة التحكم بعزم وسرعة المحركات التحريضية بمساعدة إدخال المقاومات إلى دارة الدوار لا تسمح بالحصول على تنظيم ناعم وسرعات منخفضة ومستقرة في حالة العمل القاسي جداً مما يؤدي إلى خروج الآلية عن نطاق التحكم والسيطرة.

من أجل وضع الحلول الهندسية لمشكلة العمل المتزامن لعدة محركات تقود حركة آلية المسير في الرافعة من المفضل أن تكون هذه الحلول غير محتوية على أية تجهيزات إضافية ولذلك يفضل استبعاد الحلول الميكانيكية كون الجملة الكهروميكانيكية في الرافعة ذات تركيب هيكل معقد.

وبالتالي سوف نعتمد على استخدام الحلول بالطرق الكهربائية من خلال التحكم بالفيض الموجه للمحركات التحريضية و دارات التغذية العكسية.

إن التغذية العكسية السالبة تساهم في التقليل من قيمة الخطأ عند الخرج. كما يعد تحقيق الحركة المتزامنة أمراً هاماً لعدة أسباب نذكر منها [2]:

1. يؤثر الاختلاف في المسير الناتج عن الاختلاف في سرعة المحركات على أجزاء الرافعة ويزيد من تحميلها بحمولات ديناميكية إضافية تتسبب في زيادة اهتزاز الرافعة أثناء العمل وبالتالي يقلل من دقة العمل ويؤخر العملية الإنتاجية.

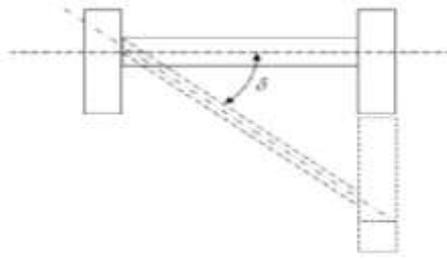
دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

2. ظهور الاختلاف بالمسير يسيء إلى جودة أداء الرافعة وخاصة في حالة رفع الأحمال المعلقة بأمراس حيث يؤدي إلى زيادة اهتزاز الحمل المعلق وتبدو هذه المشكلة بشكل أخطر إذا كانت الأحمال مواد سائلة أو معادن مصهورة.

3. يزيد الاختلاف بالمسير من صعوبة تأمين التوقف الدقيق في المكان المطلوب.

4. كل ما ذكر يساهم بشكل عام في خفض الإنتاجية الفعلية للرافعة.

ويعرف انزياح هيكل الرافعة بأنه انحراف الجسور العرضانية للهيكل بزوايا عن الوضعية الطبيعية العمودية على السكة التي تتحرك عليها الرافعة كما هو موضح في الشكل (2):



الشكل (2) انحراف المسار بين محركي الرافعة

ويظهر هذا الانزياح بسبب اختلاف القيم اللحظية للسرعات الخطية وللتسارعات الخاصة بمحاور الارتكاز في هيكل الرافعة أثناء الحركة وينتج هذا الاختلاف عن أحد الأسباب التالية:

1. الأخطاء الحاصلة في تصنيع السكة التي تسيّر عليها الرافعة أو دواليب تسيير الرافعة.

2. اختلاف وضعية المسير تحت هيكل الرافعة.

3. اختلاف نسبة تحميل المحركات بسبب اختلاف أماكن توضع الحمل المعلق بالمرس المرن.

4. عدم تناظر المميزات الميكانيكية للمحركات الكهربائية التي تقود حركة الرافعة عند محاور الارتكاز [3].

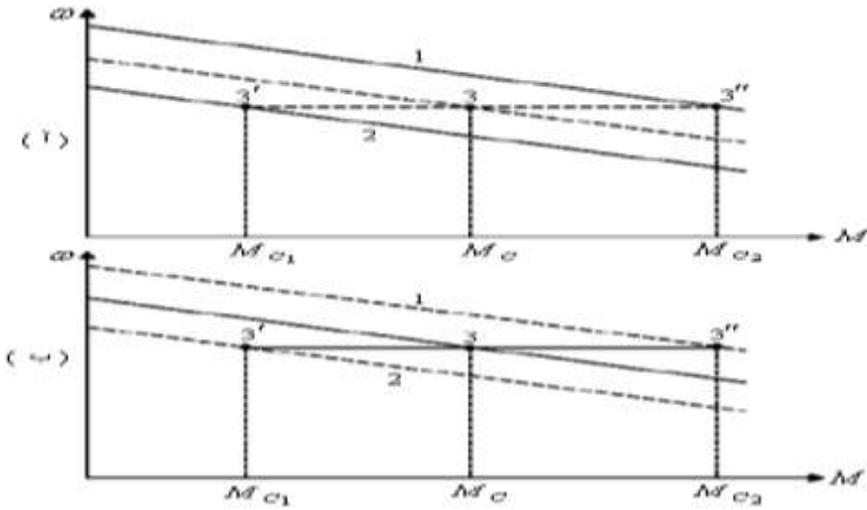
ويتسبب هذا الانزياح بنشوء حمولات إضافية في عناصر الجملة الكهروميكانيكية للرافعة تؤدي بدورها إلى تداخل في توازن القوى والعزوم المؤثرة نتيجة لعزم المرونة الناجم عن اختلاف المسير.

إن حل مشكلة الانحراف بالمسار في الجسم الهيكلي للرافعة يتحقق بالتأثير على عمل نظام العمل محرك تحريضي - مبدلة ترددية وبشكل عام إن مبدأ عمل التغذية العكسية السالبة هدفه خفض نسبة الانحراف، "طبعاً الانحراف سببه اختلاف سرعتي المحركين" ولذلك يستخدم عادةً نوعان من التغذية العكسية:

أ- تغذية عكسية بفرق سرعتين: تغذية عكسية مرنة.

ب- تغذية عكسية بفرق المسافة: تغذية عكسية قاسية.

يبين الشكل (3) المميزات الميكانيكية التي تظهر حالة التزامن في الأجزاء المتحركة للآلية.



الشكل (3) المميزات الميكانيكية التي تظهر عمل نظام التزامن

أ- من أجل جهود دخل غير متساوية $U_{Z1} \neq U_{Z2}$ ،

ب- من أجل حمولات غير متساوية على المحركات $M_{C1} \neq M_{C2}$

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

على الشكل (3) تظهر مجموعتين من المميزات الميكانيكية: مجموعة منقطة ومجموعة بخط عريض مستمر تحت تأثير نظام عمل المحرك مع المتحكم والمحددتين بالنقطتين (1,2) وبالنقاط الثلاثة (3,3',3'') بعد نهاية الحالة العابرة عند التأثير على التوالي:

- 1- إما على حساب القساوة الميكانيكية.
- 2- إما على حساب الوصلات الكهربائية في دارة التغذية العكسية للمنظم

الشكل (3-أ)

يظهر عمل الدارة مع قيم متساوية للعزم الستاتيكي على المميزات الميكانيكية ذات القساوة الميكانيكية المتساوية حالة تساوي ثوابت التضخيم في نظام الدارة المفتوحة لنظام التحكم بعمل المبدل ولكن من أجل جهود دخل مختلفة $U_{z1} > U_{z2}$ وتحديداً لذلك نقاط التشغيل هنا النقطتان (1,2).

الآن عندما يتم حذف الانحراف بالسرعتين على حساب قساوة الهيكل المعدني أي على حساب القساوة عندئذ كلا المحركين في نهاية الحالة العابرة سوف يعملان بسرعة واحدة في النقاط (3',3''). لذلك قساوة الهيكل المعدني توازن عمل المحركات على حساب إعادة توزيع الحمل بين محوري المحركين ولكن دون أي تغير في الميزة الميكانيكية لكل منهما وهي على حساب انتقال نقاط العمل من وضعية أولى إلى وضعية ثانية $2M_c = M_{c1} + M_{c2}$. ولكن عند تأثير فقط وصلات التغذية العكسية (دارات المنظم) فإن كلا المحركين يدوران بسرعات متساوية والممثلة بالنقطة (3) على حساب نقل المحركات إلى مميزات ميكانيكية جديدة مشابهة تماماً للميزة بالخط المنقط.

الشكل (3-ب)

يظهر نفس الصورة المبينة في (3-أ) ولكن بشروط جهود الدخل المتساوية $U_{z1} = U_{z2}$ وعدم تساوي الأحمال النقطتان (1,2). هنا أيضاً النقطة (3) الموافقة للعمل بسرعة ثابتة لكلا المحركين بعد تأمين عمل حالة التوازن التي تحقق توازن الحمولات على المحركات $M_c = 0.5(M_{c1} + M_{c2})$ ، وفي نفس الوقت أنظمة التوازن الكهربائي تقود إلى العمل مع نفس السرعة ولكن في النقطتين (3',3'') على مميزتين جديدتين.

الآن: عند العمل بنفس الوقت في التأثير على قساوة المميّزة الميكانيكية وعلى نظام التزامن (وهو عملياً ممكن) سنحقق عندئذ كلتا الحالتين:

أولاً: إعادة توزيع الأحمال، ثانياً: الانتقال للعمل على مميّزة ميكانيكية جديدة.

انطلاقاً مما تقدم تصبح مسألة تصميم طرق التحليل الرياضي ووضع النماذج الرياضية للحمولات الديناميكية في النظم الكهروميكانيكية عديدة الكتل ذات المحركات التحريضية العديدة المقادة بطرق التحكم المختلفة مع الأخذ بالحسبان أثر الوصلات المرنة بين المحرك والآلية وحالة اهتزاز الحمل المعلق إلى حبل مرن مسألة هامة جداً لا سيما عند دراسة حالات الاستقرار في شروط الجودة والاستقرار المطلوبين مسبقاً بشروط العملية التقنية.

5- طرق التحكم بالمحرك التحريضي

إحدى أشهر الخوارزميات المستخدمة في أنظمة القيادة الكهربائية ذات الحلقة المفتوحة هي خوارزمية القيادة السلمية ($V/f=ct$) يقوم المبدأ النظري للخوارزمية على الحفاظ على نسبة ثابتة بين الجهد والتردد المطبقين على المحرك، مما يعني ثبات الفيض المغناطيسي في المحرك، والمحافظة على عزم ثابت. لضمان العمل مع فيض مغناطيسي ثابت في المحرك لا بدّ من الحفاظ على نسبة ثابتة بين القوة المحركة الكهربائية العكسية والتردد، أي أن جهد التغذية الواجب تطبيقه على المحرك يجب أن يتغير وفق نسبة ثابتة مع التردد المطبق، ويتم اختيار هذه النسبة بحيث تتساوى قيمة الفيض الناتجة مع قيمة الفيض الإسمية للمحرك. تبين المعادلة (1) القانون الأساسي ($V/f=ct$).

$$\frac{V_s - R_s I_s}{2\pi f_s} = \Phi_s = ct \quad (1)$$

بإهمال هبوط الجهد على مقاومة الثابت لصغره عند السرعات العالية نسبياً فإن القوة المحركة الكهربائية العكسية يمكن تقريبها إلى قيمة الجهد المطبق، وبالمقابل عند السرعات المنخفضة تكون قيمة القوة المحركة الكهربائية العكسية صغيرة جداً بينما يكون هبوط الجهد على مقاومة الثابت ملحوظاً، لذا لا بدّ من تعويض هبوط الجهد هذا، من خلال حقن جهد إضافي لدعم المحرك عند السرعات المنخفضة لضمان فيض مغناطيسي ثابت فيه، وغالباً ما يتم اختيار قيمته بحدود (0-15%) من الجهد الإسمي للمحرك ويسمى (Boosting Voltage) [2].

أظهرت الأبحاث أن خوارزمية القيادة السلمية ($V/f = ct$) تعاني من مشاكل عدة عند السرعات العالية والمنخفضة، فعند السرعات المنخفضة يظهر أثر هبوط الجهد في مقاومة الثابت، لذلك يتم تعويض هذا الهبوط بجهد داعم كما تم شرحه سابقاً، أما في السرعات العالية فإن وجود الجهد الداعم يسبب زيادة في قيمة الجهد المغذي للمحرك مما يسبب دخول المحرك في الإشباع، وبالتالي فإن إضافة الجهد الداعم غير ضروري في السرعات العالية [5]. من أجل التخلص من هذه المشاكل ظهرت خوارزمية القيادة السلمية في نظام الحلقة المغلقة وهي أيضاً تعد من أبسط الطرق لتنظيم سرعة المحرك.

كما هو معلوم من العلاقة (1) فإن خوارزمية القيادة ($V/f = ct$) قائمة على فرضية ثبات النسبة (طويلة فيض الثابت) بين طويلة شعاع الجهد والتردد الزاوي في الثابت. يمكن إذاً بناء خوارزمية ($V/f = ct$) في نظام الحلقة المغلقة مع تنظيم كل من السرعة وطويلة شعاع فيض الثابت وبالتالي التخلص نهائياً من الجهد الداعم.

رغم عمل نظام القيادة السلمية ($V/f = ct$) في الحلقة المغلقة عند تنظيم السرعة مع إضافة جهد داعم أو من دون إضافة جهد داعم عبر تنظيم فيض الثابت مع تنظيم التيارات أو من دون تنظيمها، إلا أن أداء النظام في الحالة الديناميكية كان منخفضاً، حيث يتم الاعتماد على قانون سلمي في توليد جهود أو تيارات الثابت الثلاثية، أي أن النظام الثلاثي لا يمنح معلومة دقيقة عن طويلة شعاع الجهد وزاويته (مكان شعاع الجهد) عند مختلف ظروف التشغيل، لذلك يبقى نظام القيادة هذا محصوراً في التطبيقات ذات الأداء الديناميكي المنخفض، أما أجل التطبيقات ذات الأداء العالي فلا بد من استخدام نظام قيادة ذي أداء مرتفع في الحالتين الديناميكية والستاتيكية كنظام التحكم الشعاعي (FOC). الذي اقترح من قبل Hasse بعام 1969 و Blaschke في 1972، ويشكل، المثال الأكثر أهمية بالنظرية والتدريب على التحكم بالمحركات التحريضية. بالتعريف، الهدف من توجيه الفيض هو جعل المحرك التحريضي يحاكي محرك التيار المستمر ذو التهيج المستقل كمنبع للعزم القابل للتعديل، [6] (العزم الكهرومغناطيسي يتناسب طردياً مع مركبة التيار $i_{s\beta}$ وذلك في جملة محاور إحداثيات ثنائية (α, β) دوارة بسرعة الساحة المغناطيسية (ω_k, ω_s))، منذ ذلك الوقت واصل الباحثون تطوير هذه الخوارزمية، إلى أن أصبحت خوارزمية التحكم الشعاعي في عصرنا الحالي منتشرة بكثرة في التطبيقات الصناعية ذات الأداء المرتفع.

في هذا البحث قمنا بالاستفادة من تصميم نظام القيادة الشعاعية للمحرك التحريضي FOC الذي ساهم بتحسين أداء المحرك التحريضي (الذي قمنا بتقديمه في البحث المقبول للنشر في مجلة جامعة البعث بتاريخ 3-4-2022) في تحسين أداء النظم الكهروميكانيكية [10].

6- التمثيل الرياضي للمحرك التحريضي ثلاثي الطور و تنظيم سرعته بالاعتماد على نظام القيادة الشعاعية:

يعطى النموذج الرياضي للمحرك التحريضي ثلاثي الطور الذي يستخدم مركبتي شعاع فيض الدوار $\Phi_{r\alpha}^k, \Phi_{r\beta}^k$ مع مركبتي شعاع تيار الثابت $i_{s\alpha}^k, i_{s\beta}^k$ كمتحولات حالة، في نظام محاور الإحداثيات الثنائية المتعامدة والدوارة بسرعة عشوائية ω_k كما يلي [8]:

$$\begin{bmatrix} \dot{i}_{s\alpha}^k \\ \dot{i}_{s\beta}^k \\ \dot{\Phi}_{r\alpha}^k \\ \dot{\Phi}_{r\beta}^k \end{bmatrix} = \begin{bmatrix} -a_5 & \omega_k & a_3 & a_4\omega \\ -\omega_k & -a_5 & -a_4\omega & a_3 \\ a_2 & 0 & -a_1 & \omega_k - \omega \\ 0 & a_2 & -\omega_k + \omega & -a_1 \end{bmatrix} \begin{bmatrix} i_{s\alpha}^k \\ i_{s\beta}^k \\ \Phi_{r\alpha}^k \\ \Phi_{r\beta}^k \end{bmatrix} + \begin{bmatrix} b & 0 \\ 0 & b \\ 0 & 0 \\ 0 & 0 \end{bmatrix} \begin{bmatrix} v_{s\alpha}^k \\ v_{s\beta}^k \end{bmatrix} \quad (2)$$

حيث أن ثوابت المعادلة السابقة تعطى بالعلاقات التالية:

$$a_1 = \frac{R_r}{L_r}, \quad a_2 = \frac{L_m R_r}{L_r}, \quad a_3 = \frac{L_m R_r}{\sigma L_s L_r^2}, \quad a_4 = \frac{L_m}{\sigma L_s L_r},$$

$$a_5 = \frac{L_r^2 R_s + L_m^2 R_r}{\sigma L_s L_r^2}, \quad b = \frac{1}{\sigma L_s}$$

أما علاقة العزم الكهرومغناطيسي فيمكن كتابتها بدلالة مركبات التيار في الجزء الثابت ومركبات فيض الجزء الدوار كما يلي:

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

$$T_{em} = \frac{PL_m}{L_r} (\Phi_{r\alpha}^k i_{s\beta}^k - \Phi_{r\beta}^k i_{s\alpha}^k) \quad (3)$$

والمعادلة الميكانيكية يعبر عنها بالعلاقة التالية:

$$\frac{d\omega}{dt} = \frac{p}{J} (T_{em} - T_d) - \frac{f}{J} \omega \quad (4)$$

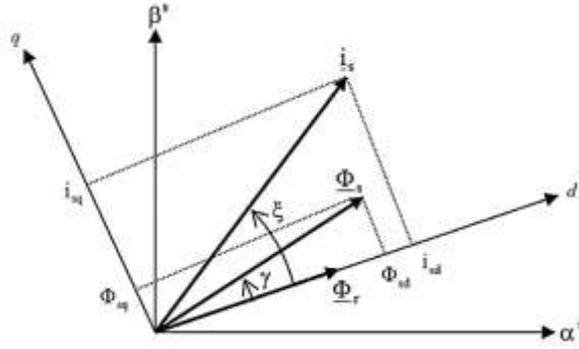
حيث $v_{s\beta}^k, v_{s\alpha}^k$: مركبتا شعاع جهد الثابت في نظام محاور إحداثيات ثنائية متعامدة ودوارة
بسرعة عشوائية ω_k .

$\omega = p\Omega$: السرعة الزاوية الكهربائية للدوار وتساوي حاصل ضرب السرعة الزاوية الميكانيكية
للدوار Ω بعدد أزواج الأقطاب p ، L_m : المحارضة المتبادلة للثابت والدوار، L_s : المحارضة
للثابت، L_r : المحارضة للدوار، $\sigma = 1 - L_m^2 / (L_s L_r)$ وتعرف بعامل التشتت، J : عزم
العطالة، T_d عزم الحمل المقاوم، f : ثابت عزم الاحتكاك اللزج.

في المعادلتين (2) و (3) المقادير الكهربائية (التيارات، الفيوض والجهود) في الثابت والدوار التي
تم التعبير عنها في نظام محاور الإحداثيات المتعامدة α^k, β^k والدوارة بسرعة عشوائية ω_k .

في الواقع، بالاعتماد على نظام القيادة الشعاعية، يمكن تقريب أداء المحرك التحريضي ثلاثي
الطور بمحرك تيار مستمر ذي تهيج مستقل حيث يمكن تمثيله رياضياً وذلك باعتبار نظام
محاور الإحداثيات الثنائية دوار بسرعة الحقل الدوار، أي $\omega_k = \omega_s$ ، وتوجيه شعاع فيض الدوار
 $\underline{\Phi}_r$ باتجاه المحور α كما هو مبين في الشكل (4) أي أن:

$$\begin{aligned} \Phi_r &= \Phi_{r\alpha} = \Phi_r \\ \Phi_{r\beta} &= 0 \end{aligned} \quad (5)$$



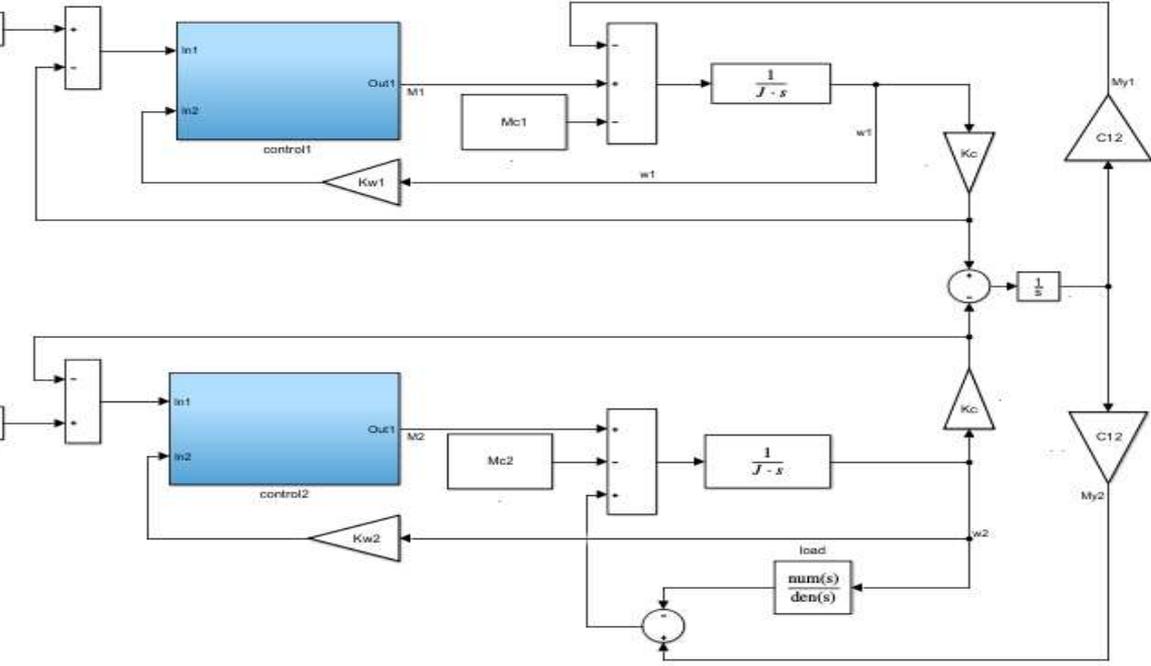
الشكل (4): توجيه شعاع فيض الدوار باتجاه المحور d

7- محاكاة نموذج النظام الكهروميكانيكي ثنائي الكتل، ثنائي المحركات التحريضية المقادة شعاعياً:

انطلاقاً من النموذج الرياضي للتحكم بالفيض الموجه بمحرك تحريضي ثلاثي الطور قمنا ببناء المخطط الصندوقي لقيادة نظام كهروميكانيكي ثنائي الكتل - ثنائي المحركات التحريضية وهو عبارة عن رافعة رصيف مرفئية جسرية وقمنا بمناقشة حالات تحميل مختلفة وذلك في بيئة ماتلاب/ سيمولينك.

الشكل (5) يوضح المخطط الصندوقي الممثل للرافعة الجسرية:

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي



الشكل (5) المخطط الصندوقي الممثل للرافعة الجسرية

حيث: $M_{c1} = M_{c2} = 162.8 [N.m]$ عزم المقاومة الستاتيكي

$C_{12} = 3677.4 [N.m / rad]$ القساوة الميكانيكية منقولة لمحور المحرك.

K_{ω} ثابت التغذية العكسية بالسرعة

$$J = \frac{J_M}{m} : \text{عزم عطالة الآلية، } m \text{ عدد المحركات.}$$

بلوكة Load تعبر عن الحمل مع مرس التعليق حيث تمثل عملية اهتزاز وتأرجح الحمل بمعادلة تفاضلية غير متجانسة من الدرجة الثانية

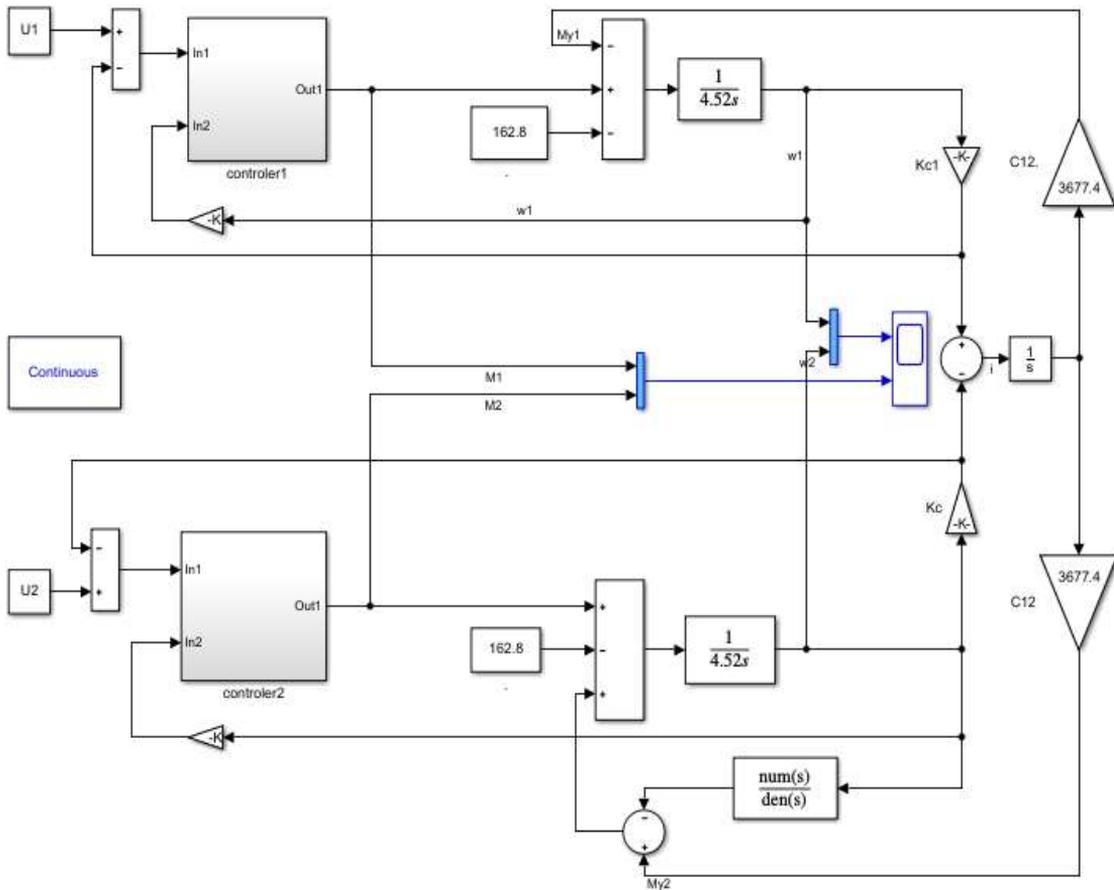
$$H_{load}(s) = \frac{M_{load}(s)}{\omega(s)} = \frac{J_{load}(s)}{T^2 s^2 + 1} \quad (6)$$

وفي الجدول (1) نبين مواصفات الحمولة التي يقودها المحرك.

الجدول (1) مواصفات الحمولة التي يقودها المحرك [9]:

الحمولة	وزن الجسم الدوار	سرعة الانتقال	نسبة التخفيض	التسارع المسموح به	قطر دولاب العربة	عدد المحركات
15T	76.5T	1.42m / sec	$i = 24.9$	$0.5m / sec^2$	0.71m	2-IM

ومنه وبعد تعويض القيم المحسوبة والتصميمية للرافعة المدروسة نحصل على الشكل (6) الذي يوضح المخطط التنفيذي لها مع الراسم المستخدم لإظهار تغيرات كل من سرعات وعزوم المحركين المستخدمين:

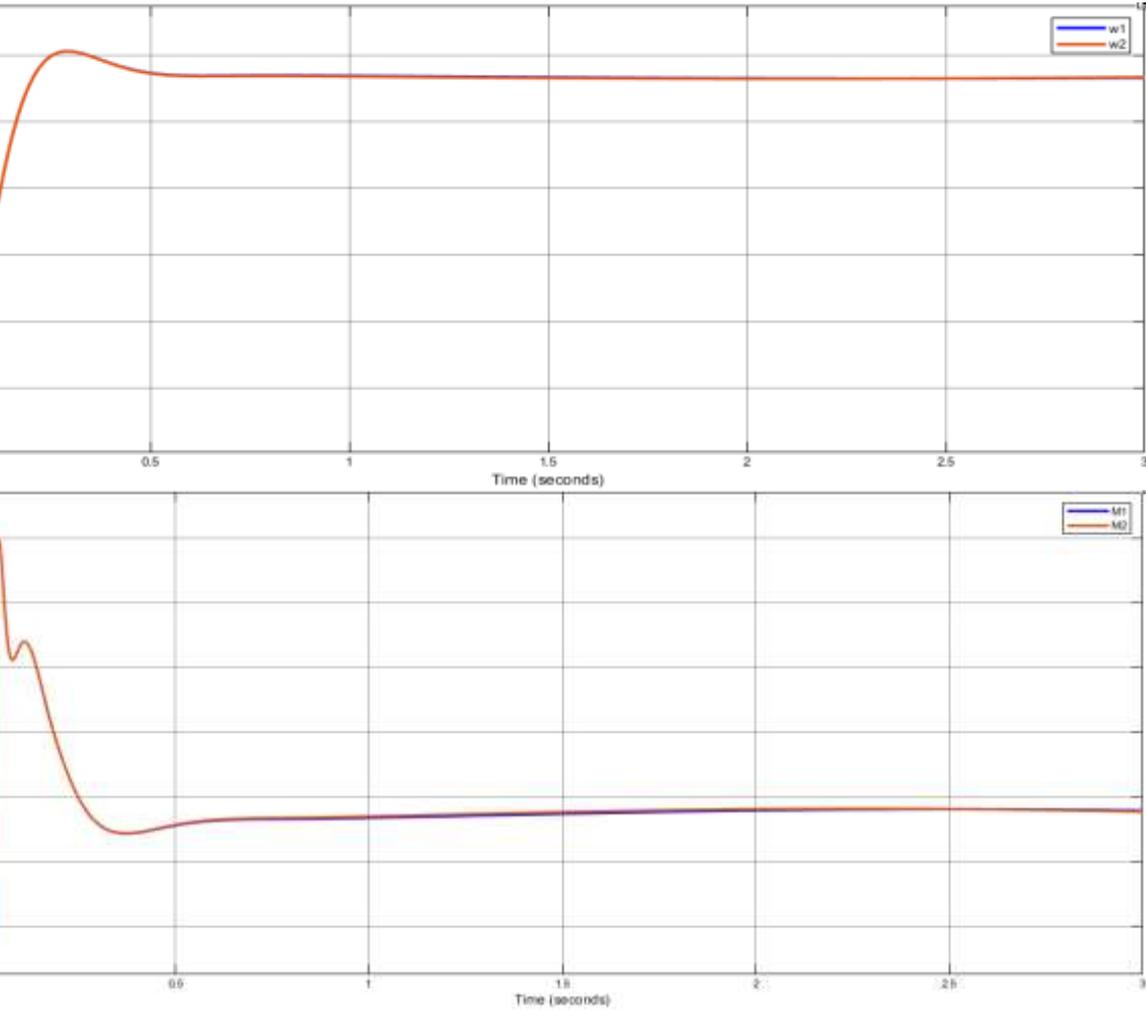


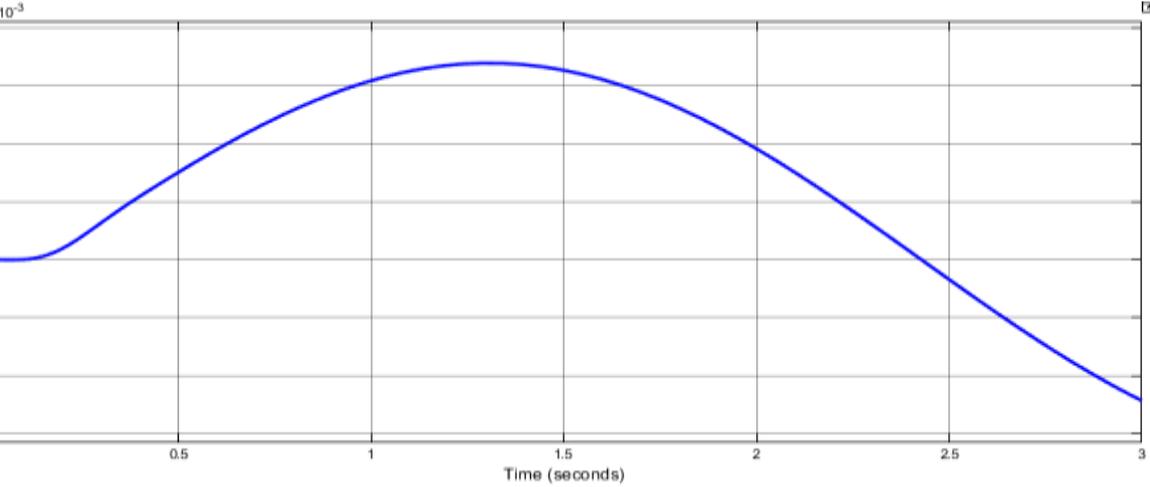
دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

الشكل (6) المخطط الصندوقي للتنفيذي للرافعة عديدة الكتل عديدة المحركات المقادة باستخدام
التحكم الشعاعي

من خلال المخطط التنفيذي السابق سنقوم بمناقشة تسع حالات لتوزع الحمولة على المحركين أولاً
بشكل متساوي ومن ثم بشكل غير متساوي وبنسب مختلفة كما يلي:

أولاً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0$ أي الحمولة موزعة بالتساوي على المحركين
التحريضيين المستقلين، يبين الشكل (7) تغير سرع وعزوم المحركين وتغير إشارة فرق المسير:





الشكل (7) النتائج عند توزيع الحمولة بالتساوي

أ- ω_1, ω_2 سرعة المحرك التحريضي الأول والثاني على التوالي

ب- M_1, M_2 العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي

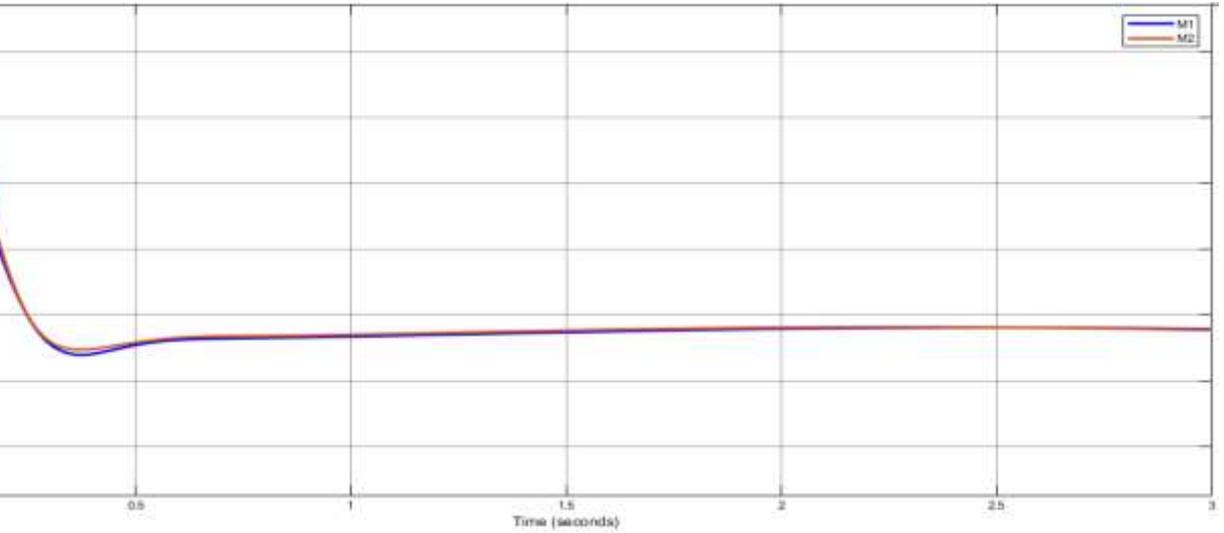
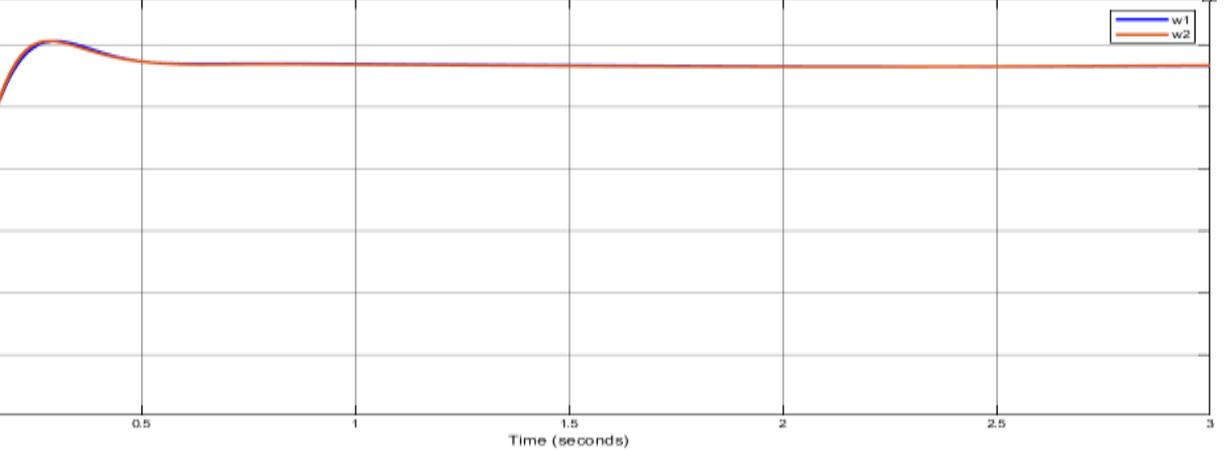
ج- إشارة الخطأ بالمسار $\Delta \varphi$

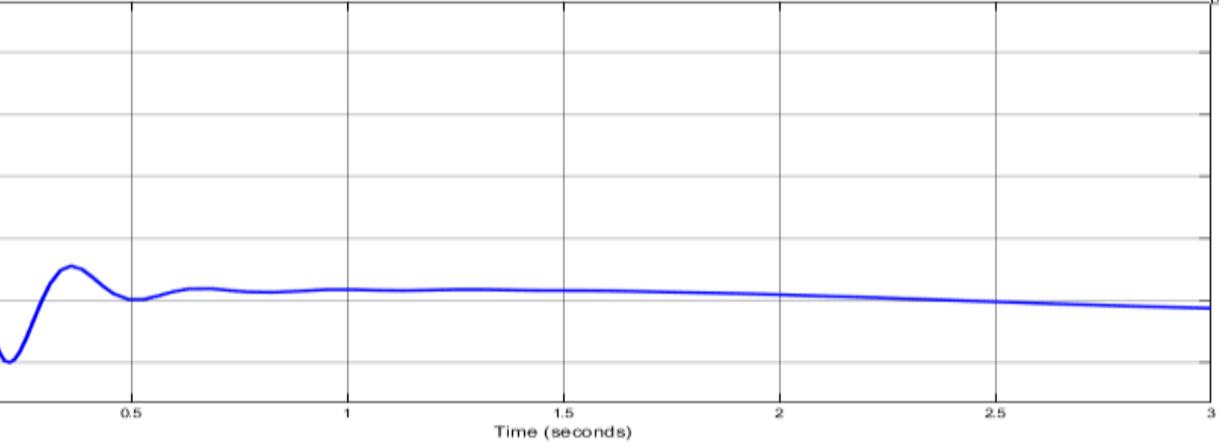
في هذه الحالة نلاحظ:

- انطباق منحنيات سرعتي المحركين وكذلك منحنيات العزم الخاصة بهما بسبب توزيع الحمولة بشكل متساوي على المحركين
- انسيابية هذه المنحنيات وعدم ظهور أي تأرجحات بالسرعة
- العزم لم يتجاوز القيمة الأعظمية لعزم المحرك والمحددة بالمواصفات الإسمية له بـ $638 [N.m]$
- زمن الحالة العابرة يساوي لـ $0.6 [sec]$.
- الخطأ بالمسار يأخذ قيم صغيرة جداً من رتبة الميلي راديان أكبر قيمة له $.3.4 \times 10^{-3} [rad]$

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

ثانياً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.2$ نلاحظ في الشكل (8) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 40% من الحمل الكامل والمحرك الثاني بنسبة 60% من الحمل الكامل.





الشكل (8) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.2$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

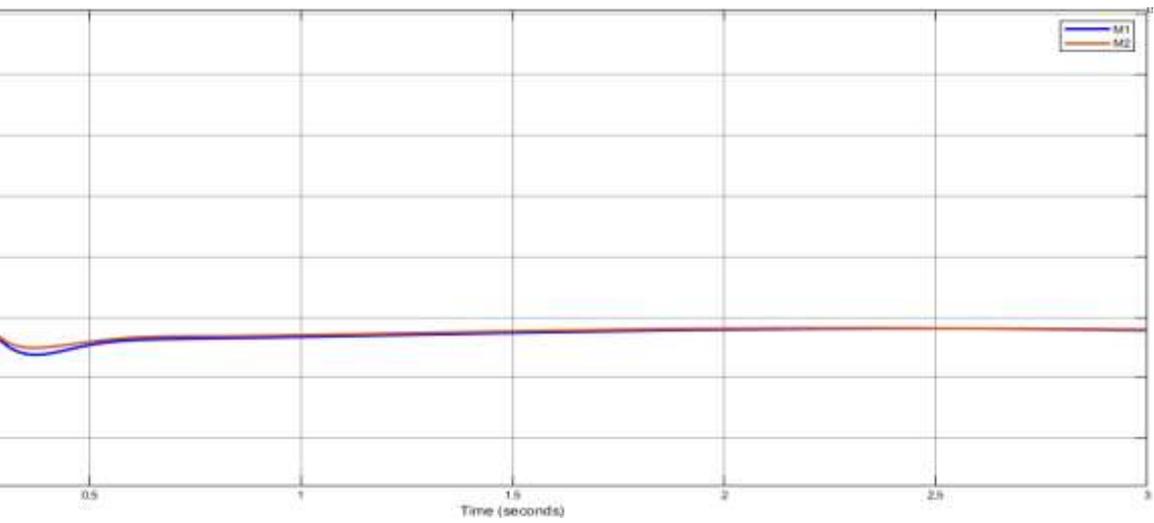
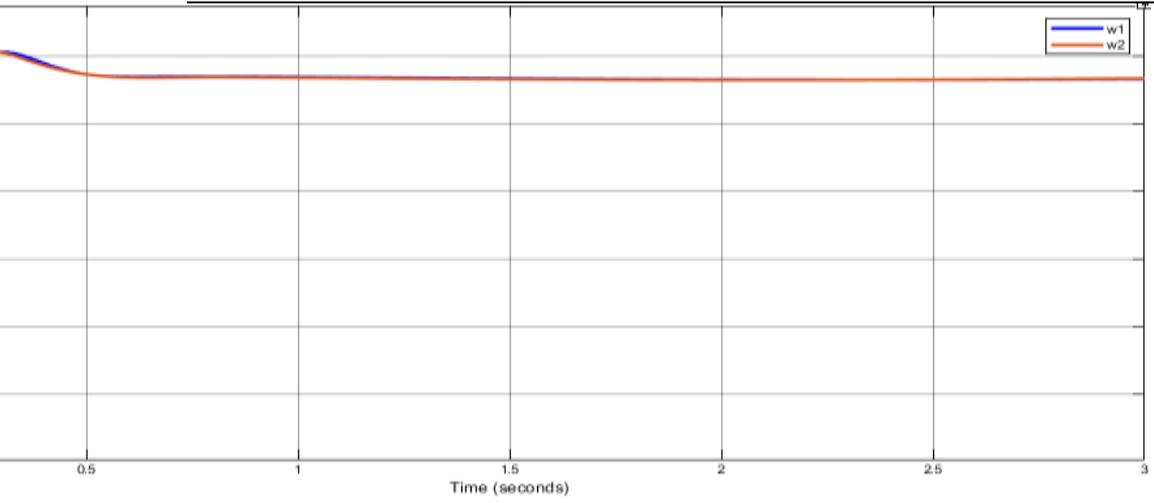
ج- إشارة الخطأ بالمسار $\Delta \varphi$

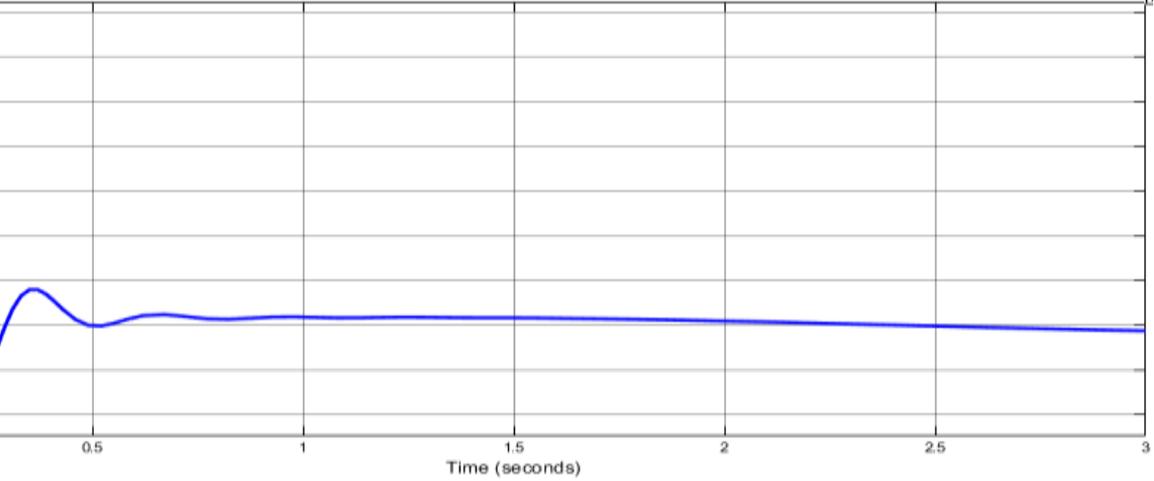
في هذه الحالة نلاحظ:

- انزياح بسيط بين منحنيني سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.
- انسيابية هذه المنحنيات وعدم ظهور أي تأرجحات بالسرعة
- العزم لكلا المحركين لم يتجاوز القيمة الأعظمية لعزم المحرك والمحددة بالموصفات الإسمية له بـ $638 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6 [sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $83.5 \times 10^{-3} [rad]$

ثالثاً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.3$ نلاحظ في الشكل (9) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 35% من الحمل الكامل والمحرك الثاني بنسبة 65% من الحمل الكامل.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي





الشكل (9) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.3$

أ- ω_1, ω_2 سرعة المحرك التحريضي الأول والثاني على التوالي

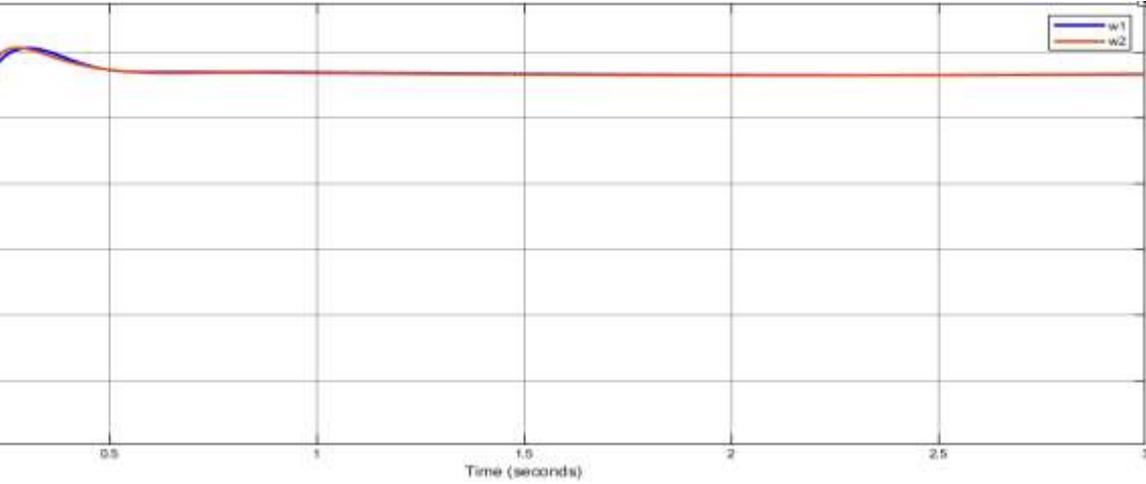
ب- M_1, M_2 العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي

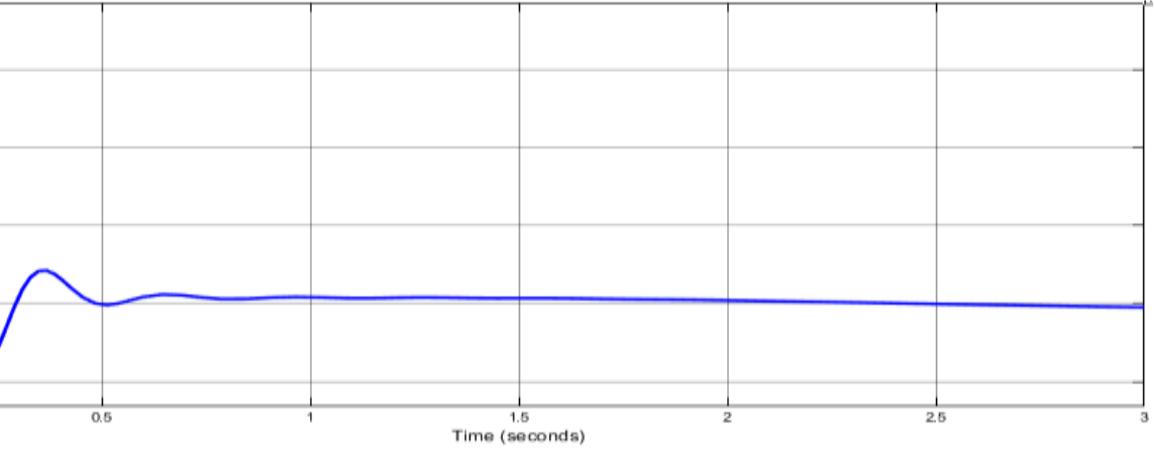
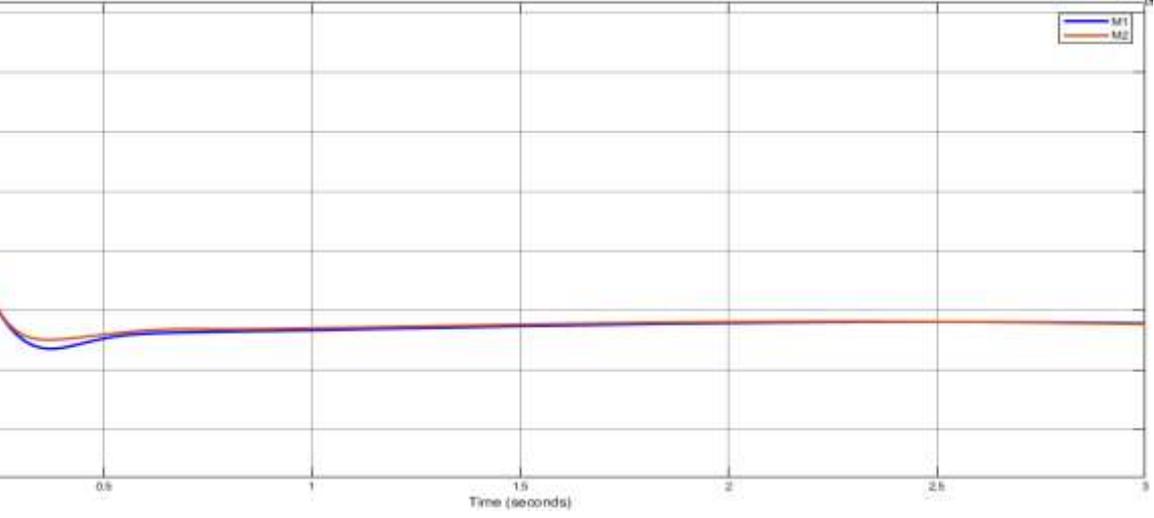
ج- إشارة الخطأ بالمسار $\Delta \varphi$

في هذه الحالة نلاحظ:

- انزياح أوضح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.
- العزم لكلا المحركين لم يتجاوز القيمة الأعظمية لعزم المحرك والمحددة بالموصفات الإسمية له بـ $638 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $125 \times 10^{-3} [rad]$

رابعاً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.4$ نلاحظ في الشكل (10) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 30% من الحمل الكامل والمحرك الثاني بنسبة 70% من الحمل الكامل.





الشكل (10) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.4$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

ج- إشارة الخطأ بالمسار $\Delta \varphi$

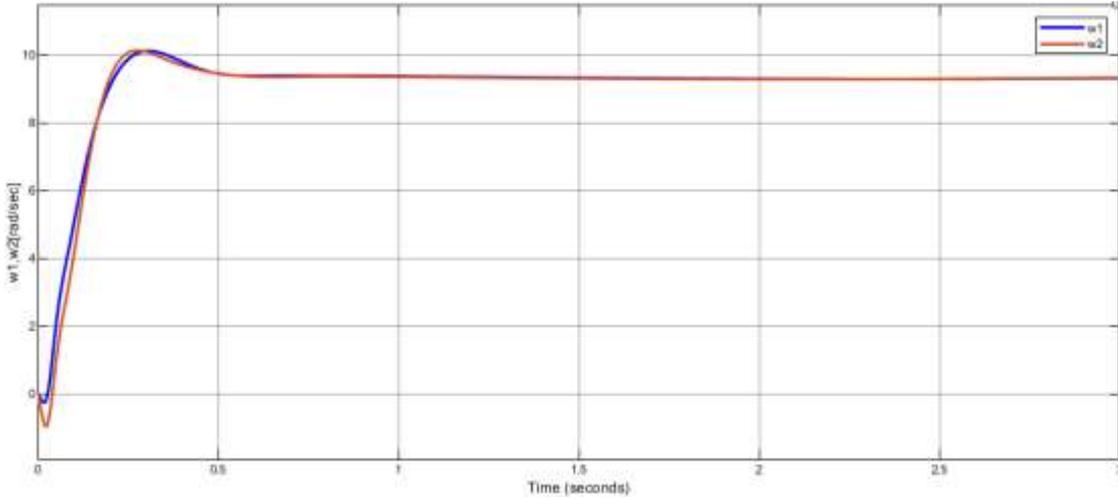
في هذه الحالة نلاحظ:

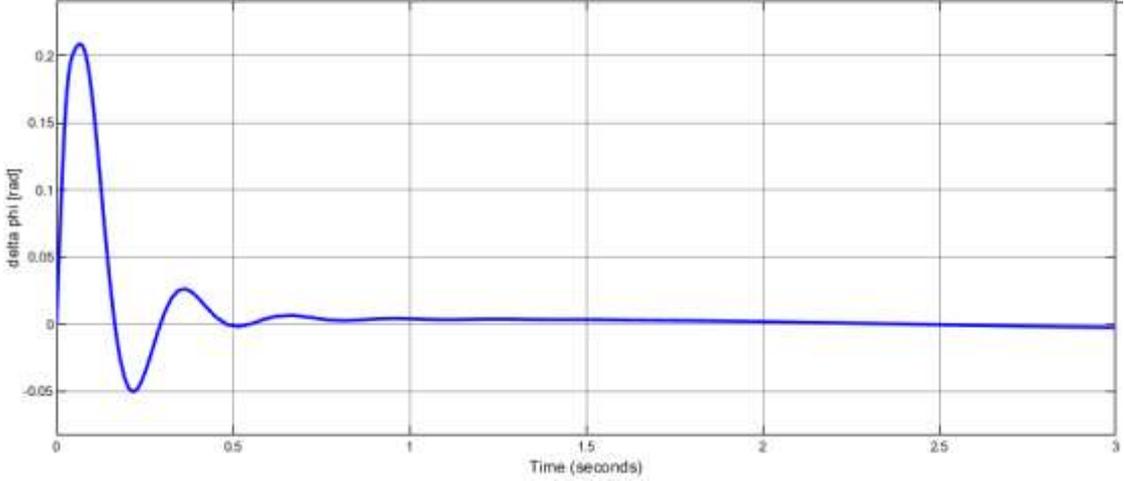
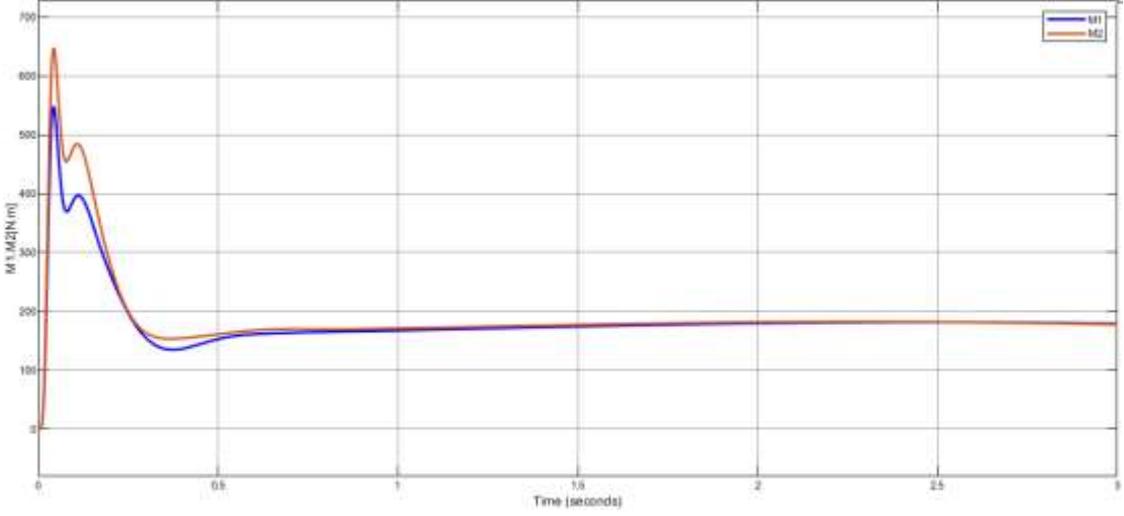
- انزياح أوضح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

- العزم لكلا المحركين لم يتجاوز القيمة الأعظمية لعزم المحرك والمحددة بالموصفات الإسمية له بـ $638 [N.m]$ ولكنه اقترب منها للمحرك الثاني (المحمل بنسبة أكبر) حيث وصل لقيمة أعظمية $637 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $.125 \times 10^{-3} [rad]$

خامساً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.5$ نلاحظ في الشكل (11) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 25% من الحمل الكامل والمحرك الثاني بنسبة 75% من الحمل الكامل.





الشكل (11) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.5$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

ج- إشارة الخطأ بالمسار $\Delta \varphi$

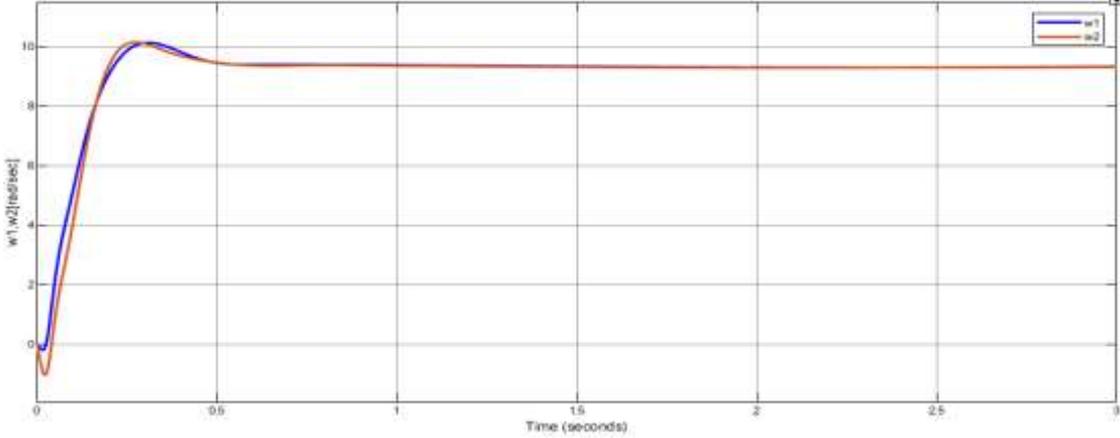
في هذه الحالة نلاحظ:

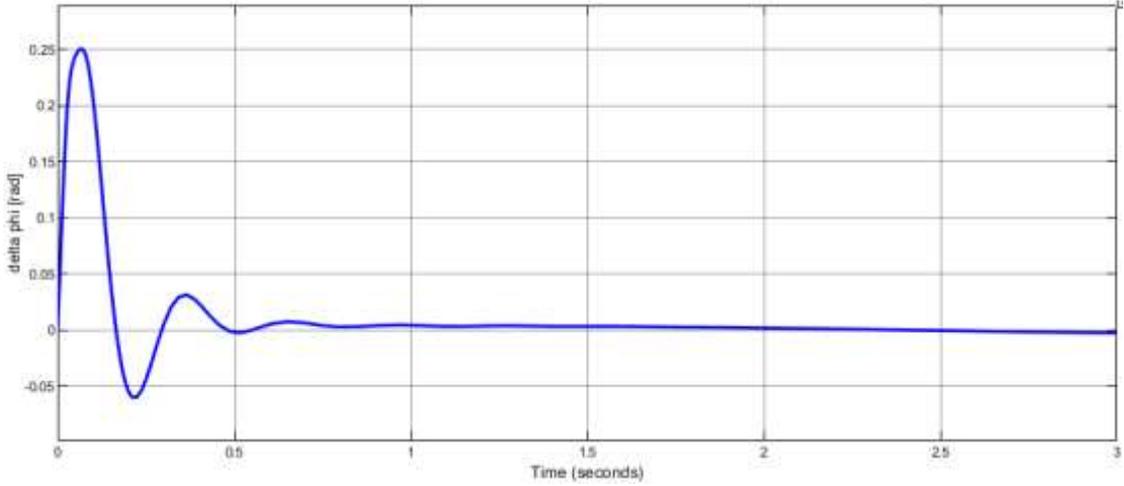
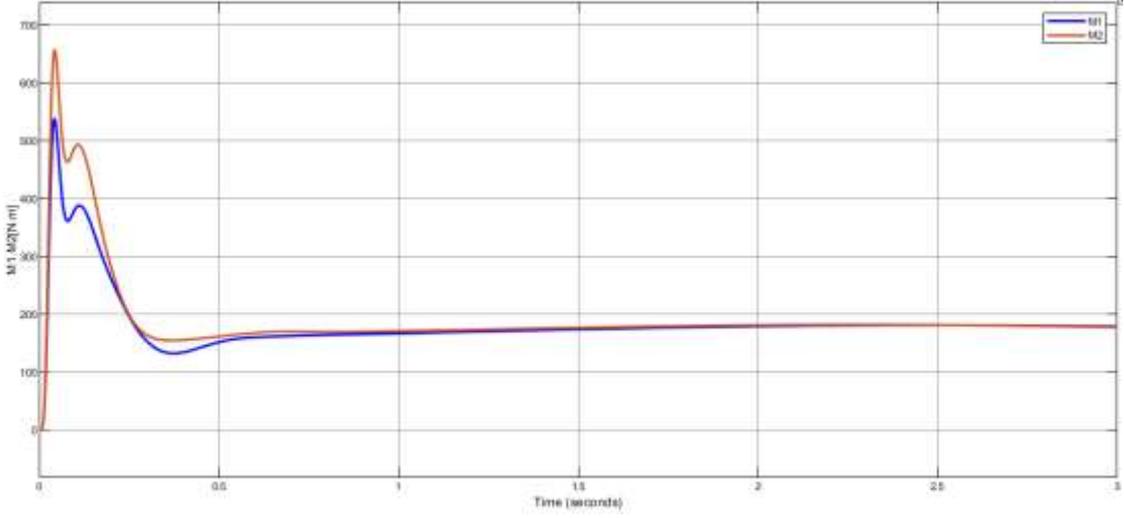
- زيادة انزياح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

- العزم للمحرك الثاني (المحمل بالنسبة الأكبر) يتجاوز القيمة الأعظمية لعزم المحرك المحددة بالموصفات الإسمية له بـ $638 [N.m]$ حيث يصل لـ $647.58 [N.m]$ ، بينما عزم المحرك الأول (المحمل بالنسبة الأصغر) يأخذ قيمة عظمى تساوي $547.3 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $208 \times 10^{-3} [rad]$

سادساً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.6$ نلاحظ في الشكل (12) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 20% من الحمل الكامل والمحرك الثاني بنسبة 80% من الحمل الكامل.





الشكل (12) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.6$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

ج- إشارة الخطأ بالمسار $\Delta \varphi$

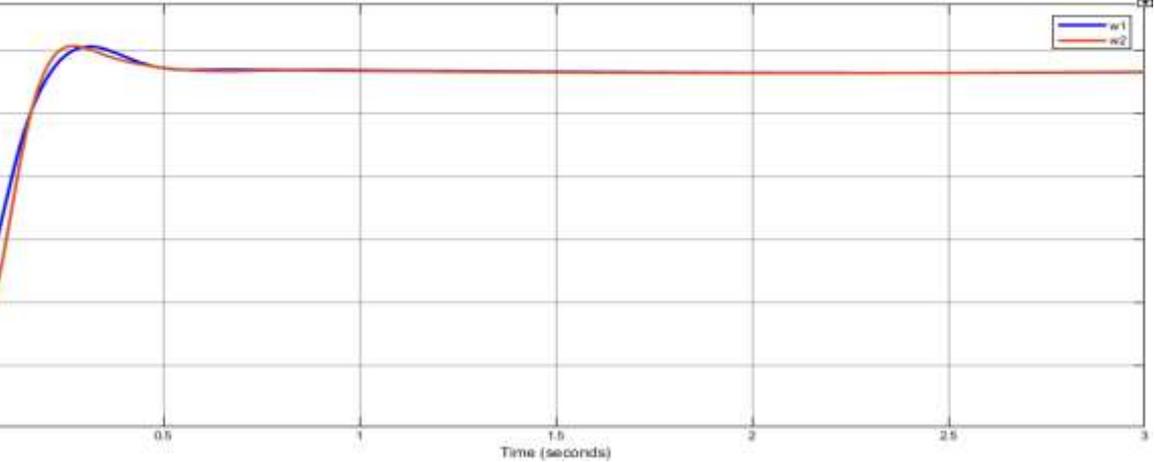
في هذه الحالة نلاحظ:

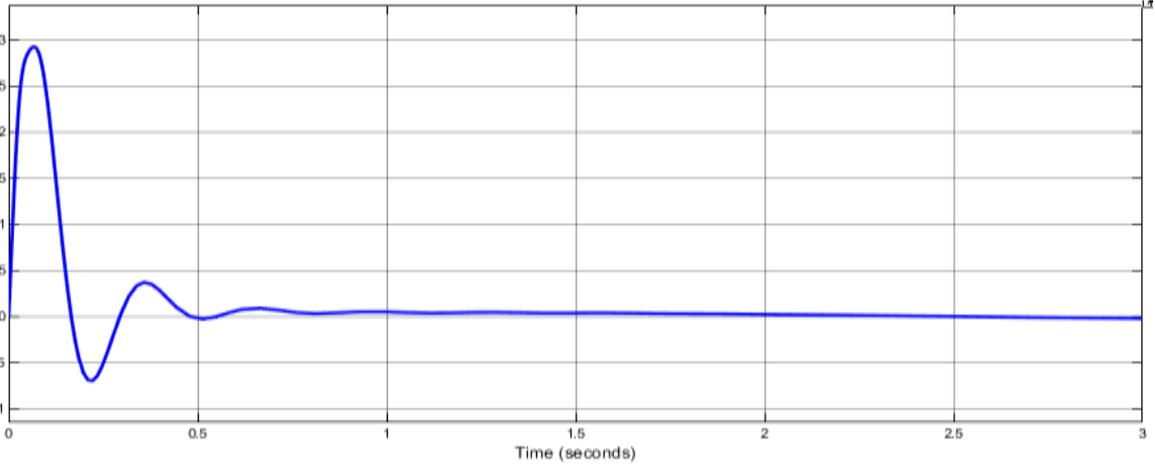
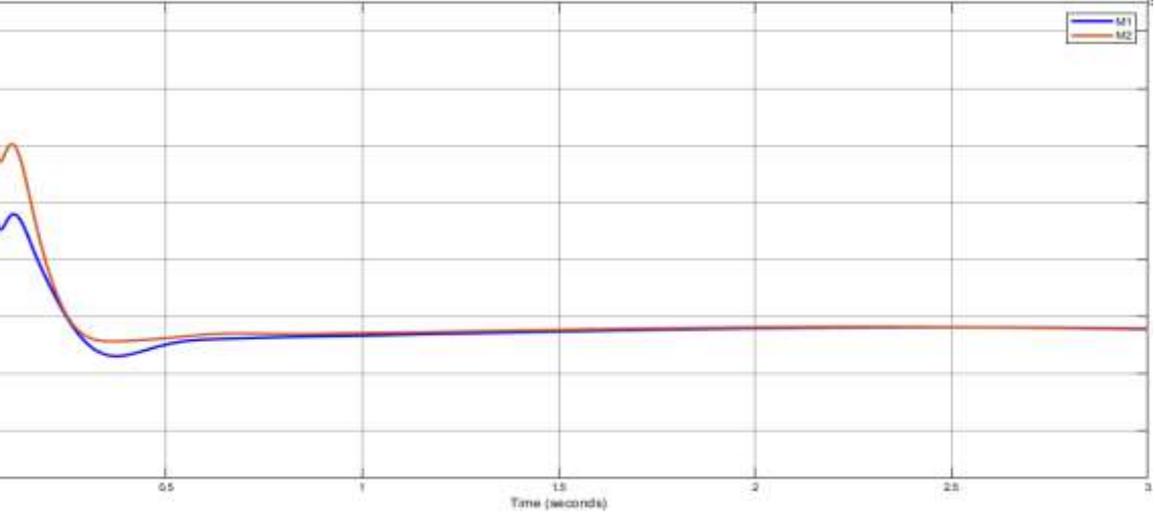
- زيادة انزياح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

- العزم للمحرك الثاني (المحمل بالنسبة الأكبر) يتجاوز القيمة الأعظمية لعزم المحرك المحددة بالمواصفات الإسمية له بـ $638 [N.m]$ حيث يصل لـ $657.11 [N.m]$ ، بينما عزم المحرك الأول (المحمل بالنسبة الأصغر) يأخذ قيمة عظمى تساوي $537 [N.m]$ فقط.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $.250 \times 10^{-3} [rad]$

سابعاً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.7$ نلاحظ في الشكل (13) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 15% من الحمل الكامل والمحرك الثاني بنسبة 85% من الحمل الكامل.





الشكل (13) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.7$

أ- ω_1, ω_2 سرعة المحرك التحريضي الأول والثاني على التوالي

ب- M_1, M_2 العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي

ج- إشارة الخطأ بالمسار $\Delta \varphi$

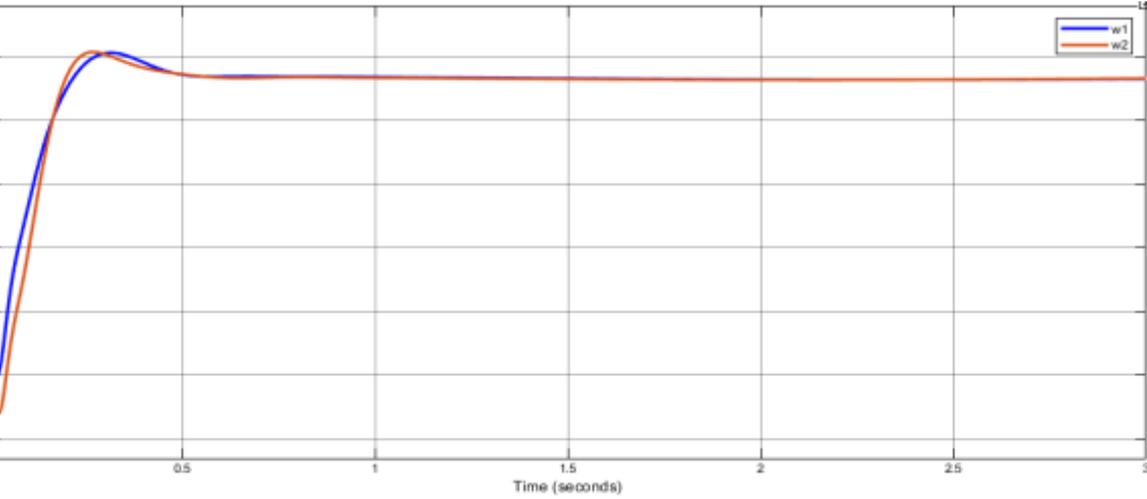
في هذه الحالة نلاحظ:

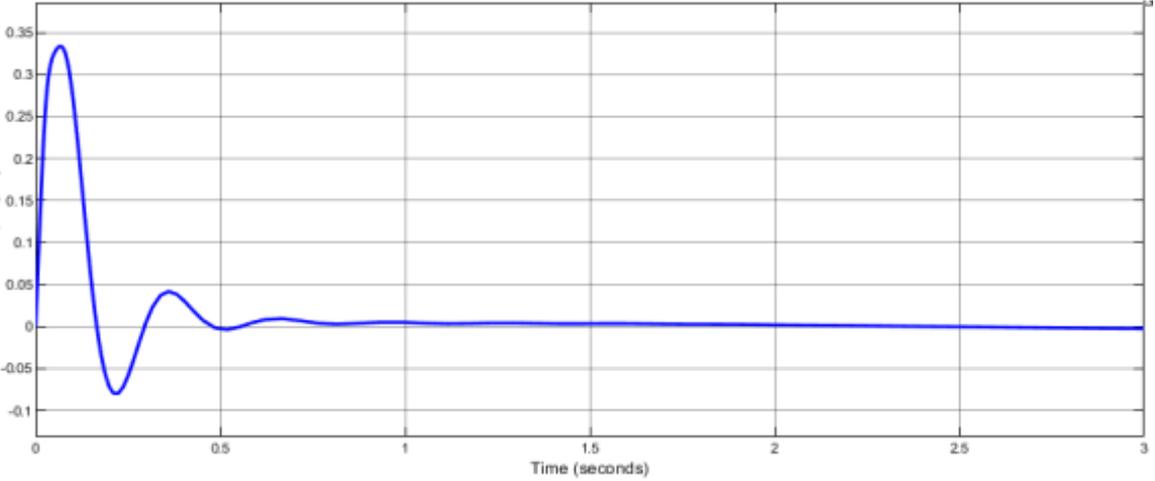
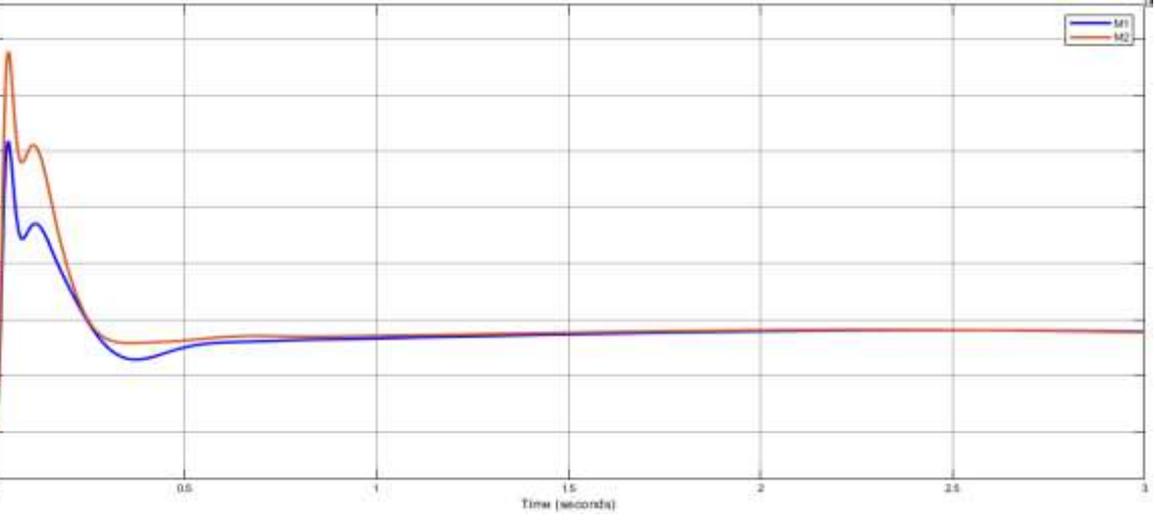
- زيادة انزياح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

- العزم للمحرك الثاني (المحمل بالنسبة الأكبر) يتجاوز القيمة الأعظمية لعزم المحرك المحددة بالموصفات الإسمية له بـ $638 [N.m]$ حيث يصل لـ $667.35 [N.m]$ ، بينما عزم المحرك الأول (المحمل بالنسبة الأصغر) يأخذ قيمة عظمى تساوي $528 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $.292 \times 10^{-3} [rad]$

ثامناً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.8$ نلاحظ في الشكل (14) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 10% من الحمل الكامل والمحرك الثاني بنسبة 90% من الحمل الكامل.





الشكل (14) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.8$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

ج- إشارة الخطأ بالمسار $\Delta \varphi$

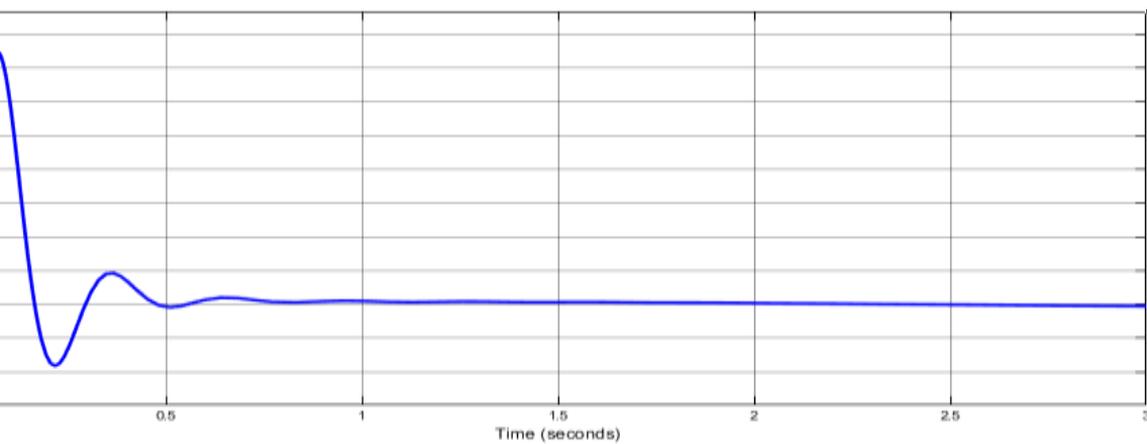
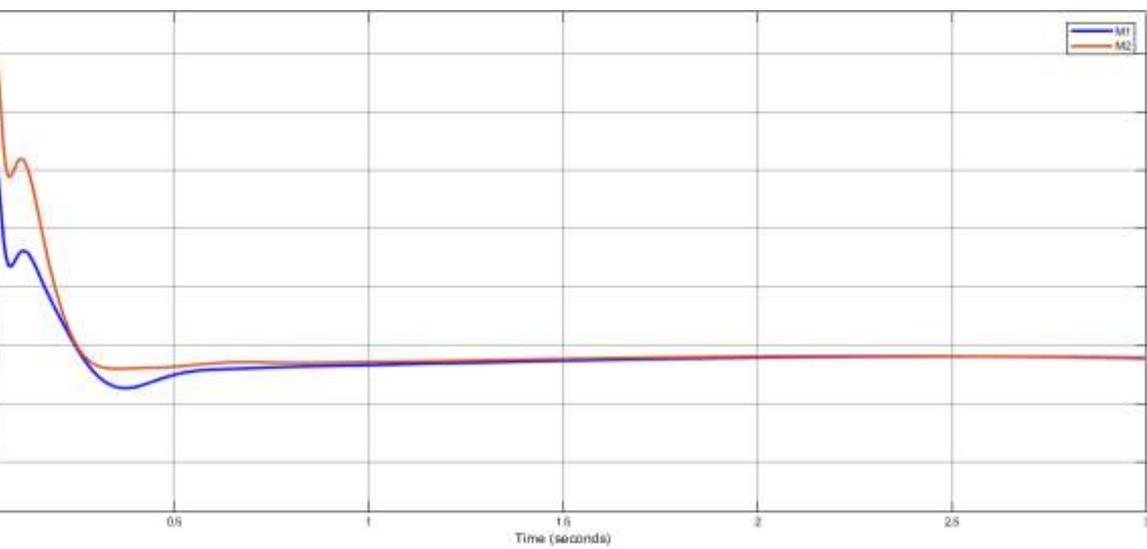
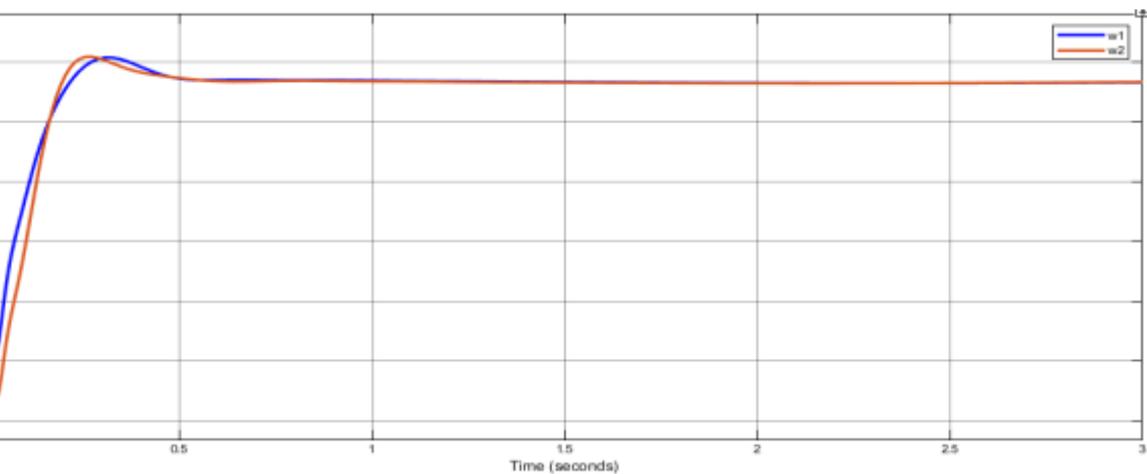
في هذه الحالة نلاحظ:

- زيادة انزياح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.

دراسة تزامن المحركات التحريضية المستقلة في النظم الكهروميكانيكية عديدة الكتل عديدة المحركات
المقادة باستخدام التحكم الشعاعي

- العزم للمحرك الثاني (المحمل بالنسبة الأكبر) يتجاوز القيمة الأعظمية لعزم المحرك المحددة بالمواصفات الإسمية له بـ $638 [N.m]$ حيث يصل لـ $677 [N.m]$ ، بينما عزم المحرك الأول (المحمل بالنسبة الأصغر) يأخذ قيمة عظمى تساوي $518 [N.m]$.
- زمن الحالة العابرة يساوي لـ $0.6[sec]$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $.334 \times 10^{-3} [rad]$.

تاسعاً: الفرق بنسب توزيع الحمولة $\Delta M^* = 0.9$ نلاحظ في الشكل (15) منحنيات السرعة والعزوم للمحركات إضافة إلى إشارة الخطأ بالمسار حيث يكون المحرك الأول محملاً بنسبة 5% من الحمل الكامل والمحرك الثاني بنسبة 95% من الحمل الكامل.



الشكل (15) النتائج عند فرق بتوزيع الحمولة $\Delta M^* = 0.9$

أ- سرعة المحرك التحريضي الأول والثاني على التوالي ω_1, ω_2

ب- العزم الكهرومغناطيسي للمحرك التحريضي الأول والثاني على التوالي M_1, M_2

ج- إشارة الخطأ بالمسار $\Delta \varphi$

في هذه الحالة نلاحظ:

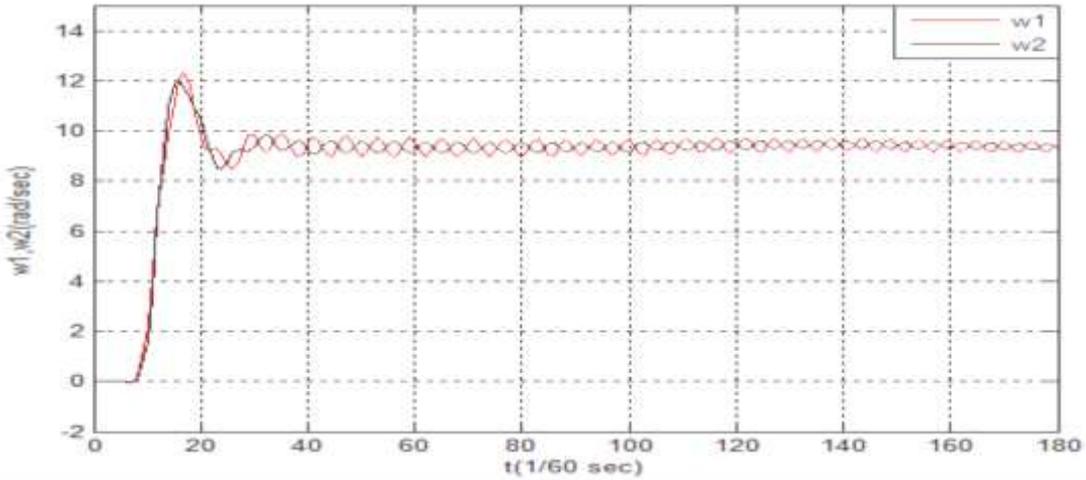
- زيادة انزياح بين منحنىي سرعتي المحركين خلال الحالة العابرة ثم في يعودان للانطباق بالحالة المستقرة.
- العزم للمحرك الثاني (المحمل بالنسبة الأكبر) يتجاوز القيمة الأعظمية لعزم المحرك المحددة بالمواصفات الإسمية له بـ $[N.m] 638$ حيث يصل لـ $[N.m] 686.7$ ، بينما عزم المحرك الأول (المحمل بالنسبة الأصغر) يأخذ قيمة عظمى تساوي $[N.m] 508$.
- زمن الحالة العابرة يساوي لـ $[sec] 0.6$.
- الخطأ بالمسار يأخذ قيم أكبر بقليل مقارنة مع الحالة السابقة حيث يصل كقيمة عظمى لـ $[rad] 375 \times 10^{-3}$.

نستنتج من الحالات التسع المدروسة لتغير نسب التحميل على المحركين المستخدمين أنه كلما ازداد الفرق بنسبة التحميل يزداد الفرق بين منحنىي سرعتين خلال حالة العابرة ومن ثم يدخلان في الانطباق والاستقرار عند زمن استقرار $[sec] 0.6$ لم يتغير في كل الحالات وهذه ميزة هامة جداً ناتجة عن استخدام طريقة التحكم الشعاعي في قيادة المحركين المستقلين.

كما نرى من خلال منحنيات العزم الناتجة أن المحرك المحمل بنسبة أكبر يعاني من هزة خلال المرحلة العابرة وهي ضمن القيم المسموح بها حتى عند الحالة التاسعة (التي تمثل أفسى حالة عمل في الرافعة) ومن ثم يستقر العزم على القيمة المطلوبة.

نلاحظ من كافة النتائج بمختلف الحالات المدروسة القيم المنخفضة جداً لفرق المسار .

يبين الشكل (16) تغير سرعتي المحركين المستخدمين في الرافعة الجسرية المدروسة باستخدام طريقة التحكم السلمي عند توزع الحمولة بالتساوي على المحركين وفقاً للمرجع [9]:



الشكل (16) ω_1, ω_2 سرعة المحرك التحريضي الأول والثاني على التوالي، عند فرق بتوزع

الحمولة $\Delta M^* = 0$ بحالة التحكم السلمي ومع استخدام منظم تناسبي $K_p=5$

بمقارنة الشكل (16) مع الأشكال (7-أ) و (8-أ) و (9-أ) و (10-أ) و (11-أ) و (12-أ) و (13-أ) و (14-أ) و (15-أ) وهي المنحنيات الممثلة لسرعتي المحركين التحريضيين المستخدمين في الرافعة المدروسة نلاحظ أن التحكم الشعاعي حقق سرعة استجابة عالية جداً واستقرار ملحوظ في السرعة على الرغم من تغير توزع الحمولة الملقاة على عاتق كل من المحركين كما تم الاستغناء عن المنظم التناسبي.

الخلاصة والنتائج:

توصلنا في نهاية البحث إلى النتيجة التالية:

1. أدى استخدام طريقة التحكم الشعاعي في آلية الرفع الكهروميكانيكية إلى الاستغناء عن استخدام المنظمات في دارة التغذية العكسية المعتمدة في أبحاث سابقة [2] و [9].

2. هذا الاستغناء أدى إلى سرعة استجابة عالية جداً للنظام المدروس وواضح من خلال مقارنة

أزمنة الاستقرار في كلا الحالتين وبمعدل $\frac{t_{s(Scalar)}}{t_{s(FOC)}} = \frac{2.84}{0.6} = 4.73$ [9] وهذا له دلالاته

الاقتصادية في خطوط الإنتاج عند تطبيق أنظمة التحكم.

3. نلاحظ من كافة النتائج بمختلف الحالات المدروسة القيم المنخفضة جداً لفرق المسار. وهذا يعود إلى الإمكانيات التي يحققها التحكم الشعاعي بفصل التحكم بالفيضان عن التحكم بالعزم. مما أدى إلى إلغاء استخدام منظمات PID المستخدمة في الأنظمة التقليدية المعتمدة سابقاً.
4. في هذا البحث حققنا الشروط المطلوبة بعد إلغاء دارات الـ PID وبالتالي اختصرنا من ثوابت التأخير الزمني في الدارة وهذا ما انعكس على زمن الإقلاع والدخول في الاستقرار.

المراجع العلمية:

- 1) Kluchev V.E. Optimal of electric drive with elasticity, Jornal of electrical system 1997 N:1, P 38-43.
- 2) Jaishankar Iyer. LOAD SHARING SCHEMES IN MULTIPLE INDUCTION MOTOR DRIVE APPLICATIONS USING VOLTS-PER-HERTZ CONTROL, B.E., The University of Mumbai, India, 2006.

3) Ivan Odnokopylov. Olga Galtseva. Jiren Li. Natalia Natalinova .The control system of synchronous movement of the gantry crane supports. Journal of Physics Conference Series. January 2017.

4) HOLTZ J., 1996– On the Spatial Propagation of Transient Magnetic Fields in AC Machines, IEEE Trans. on IA, 32(4), 927-936.

5) KOGA K.; UEDA R.; SONODA T.Constitution of V/f control for reducing the steady-state speed error to zero in induction motor drive system, IEEE Trans. OnIA, 28(2), 463-471, 1992.

6) Quang, N. P, Dittrich. Vector Control of Three-Phase AC Machines: System Development in the Practice. 2nd ed. Berlin: Springer (2015).

7) Abdul M, Biabani Kh, Ali S. Control of Induction Motor Drive using Space Vector PWM. International Conference on Electrical, Electrical, Electronics, and Optimization Techniques (ICEEOT) ,2016.

8) Jie Huang. Control of bridge cranes with distributed-mass payloads under windy conditions. Mechanical Systems and Signal Processing · May 2016.

9. دراسة أعدت لنيل درجة الدكتوراه في هندسة الطاقة الكهربائية. تحسين الأداء الديناميكي للنظم الكهروميكانيكية عديدة الكتل عديدة المحركات التحريضية المقادة ثايرستورياً بالطرق الكهربائية. بإشراف: د. شفيق باصيل. إعداد: م. رولا ميهوب. جامعة البعث. 2016-2017.

10 - دراسة أداء النظم الكهروميكانيكية عديدة الكتل - عديدة المحركات التحريضية باستخدام طريقة التحكم الشعاعي بالفيض. بإشراف أ. د. شفيق باصيل، أ. د. بسام عطيه. إعداد: م. رغبة الابراهيم. مجلة جامعة البعث. مجلد 44 لعام 2022.

إطار عمل لتخمين جهود فريق العمل في تطوير مشاريع الويب

روان انور نصر

كلية الهندسة المعلوماتية - قسم البرمجيات - جامعة تشرين

طالبة ماجستير علوم الويب في الجامعة الافتراضية السورية

المشرف: د. محمد علي محمد

دكتور مدرس في كلية الهندسة المعلوماتية والاتصالات - الجامعة العربية الدولية

الملخص:

يخوض مجال تقدير جهود إنجاز المشاريع البرمجية مصاعب كبيرة لعدم القدرة على تحديد الجهد المبذول بشكل دقيق ليتم تخمين التكلفة الصحيحة مقارنة بالجهد المبذول، لذا ومنذ عقود تم وضع العديد من الأفكار والخوارزميات والمنهجيات التي تسعى لتقدير الجهد بأقرب شكل ممكن اعتماداً على معايير متعددة، أبرزها COCOMO التي تعتمد على عدد الأسطر البرمجية و Use Case Point التي تعتمد على جداول حالات الاستخدام التي يبنى عليها المشروع البرمجي. وبعد دراسة العديد من الطرق بما فيها النموذجين المذكورين، تم التوصل إلى فكرة جديدة تعتمد على بيانات دراسة المتطلبات والدراسة التحليلية للمشروع بشكل كامل، لتحديد باقي مراحل إنجاز المشروع البرمجي والتي هي (تصميم - تنفيذ - اختبار - نشر - صيانة)، وإدخال جميع تفاصيل البيانات التي تحدد العمل الذي سيتم إنجازه لتنفيذ المشروع بدقة، كتصميم الصفحات والواجهات وقاعدة المعطيات والأكواد البرمجية الواجب كتابتها مع تحديد حجمها، بناءً على هذه الإدخالات سيتم تحديد الوقت اللازم لإنجاز المشروع وبالتالي معرفة عدد الأشخاص اللازمين لإنجاز هذا المشروع. تمت تجربة المشروع عن طريق بناء موقع ويب يسمح

بإدخال بيانات المشروع، وتم مقارنة الناتج مع نواتج COCOMO و Use Case Point، وتبين معنا أنه بتحديد بيانات المشروع بدقة حصلنا على توفير بالوقت المخصص لإنجاز المشروع، وبالتالي توفير في الكادر والتكلفة.

- الكلمات المفتاحية: الجهد - تقدير الجهد - تقدير التكلفة - تحليل متعدد المعايير -
- عامل ضبط القيمة - عدد أسطر الكود البرمجي - نقاط حالات الاستخدام - رأي
- الخبير - التناظر التاريخي.

A Framework for Estimating Efforts in Developing Web Projects

Abstract:

The field of efforts estimating of implementation software projects is facing great difficulties due to the inability to accurately determine the effort expended in order to estimate the correct cost compared to the effort expended. Therefore, for generations, many ideas, studies, algorithms and methodologies have been developed that seek to estimate the effort as closely as possible based on multiple criteria, most notably COCOMO, which Depends on the number of lines of code, and the Use Case Point is based on the use case tables on which the software is built. After studying a lot of solutions (including the past two mentioned models) a new idea was reached that relies on the data of the requirements study and a fully analytical study of the project, to determine the remaining stages of the completion of the software project, which are (design - implementation - testing - publishing - maintenance), and entering all the details of the data that determining the work that will be done to implement the project accurately, such as designing the pages, interfaces, database, and programming codes to be written, with specifying their volumes.

The project was tested by building a website that allows entering project data, and the output was compared with the outputs of COCOMO and Use Case Point, and it was found with us that by accurately identifying the project data, we got a saving in the time allocated to completing the project, and thus saving in staff and cost.

Keywords: Effort - Effort Estimation - Cost Estimation - Multi-Criteria Analysis (MCA) - Value Adjustment Factor (VAF) - Source line of code (SLOC) - Use Case Point (UCP) - Expert Judgment - Analogy Historical.

1- مقدمة البحث

1.1 - المقدمة:

يُعرى إخفاق المشاريع البرمجية في كثير من الحالات الى التقديرات غير الملائمة، أو المُبالغ فيها لعناصر التكلفة، لذلك يجب دراسة الأساليب المتبعة التي تتصف بالعمومية لمعرفة ما إذا كانت تناسب جميع الشركات على حد سواء، وبغض النظر عن خصوصية وتفرد كل شركة، وإن كانت بعض الأساليب تراعي موضوع الخصوصية بشكل بسيط. ونقصد بالخصوصية هنا خبرة الشركة مع نمط معين من المشاريع ومستوى كفاءة عاملها، بالإضافة لسهولة تعاملها مع الزبون، فبعد دراسة خصوصية كل شركة على حدة، ربما يكون من المناسب اقتراح الآلية المناسبة لها بناءً على هذه الخصوصية، وبالتالي من الممكن أن تكون الآلية المقترحة لها لا تتلاءم تماماً مع غيرها. من هنا تبرز أهمية التقدير الملائم لتكلفة المشروع البرمجي، التي تعد من التحديات المهمة في الوقت الحاضر في مجال إدارة المشاريع البرمجية، حيث يتطلب ذلك وضع تقديرات صحيحة لجميع الجوانب، بما في ذلك من تحليل وتصميم النظام وتجزير البرمجيات واختبارها، ولاسيما فيما يتعلق بتقدير الجهد البشري اللازم لإنجاز مثل هذه الأعمال.^[1]

1.2 - المشكلة العلمية ومبررات مشروع البحث:

يعتبر تخمين جهود فريق العمل من أهم التحديات التي يعاني منها مدراء المشاريع، بسبب عدم اعتمادها على أسس محددة، ولما له من أثر كبير على تقدير التكلفة النهائية للمشروع، ففي إطار تقدير تكلفة مشروع برمجي، كيف يمكن أن نأخذ بعين الاعتبار خصوصية الشركة المطورة، وكيفية تنفيذ المشروع، من أجل الحصول على تقديرات أكثر ملائمة من تلك التي تعطيها الطرائق العامة المستخدمة في هذا المجال. ولنستطيع الإجابة على هذا السؤال الجوهرى يجب أن نركز على الفكرة الرئيسية التالية: إن مشكلة تقدير الكلفة في المشاريع البرمجية تؤول في النهاية إلى مسألة تقدير الجهد اللازم للتطوير، فهذه المسألة هي الموضوع الأساسي الذي تركز

عليه الطرائق الشائعة ، أما الانتقال من تقدير الجهد إلى تقدير تكلفة هذا الجهد فهو مرتبط بالبيئة الاقتصادية التي يجري فيها تنفيذ المشروع وخاصة أجور العاملين في مجال التطوير البرمجي.

1. 3- هدف البحث:

- 1- البحث في نقاط الضعف والمشكلات التي تواجه إدارة مشاريع الويب عموماً وخاصة فيما يتعلق بتقدير تكلفة المشروع.
- 2- مراجعة الأدوات والتقنيات والمنهجيات المتوافرة واستكشاف مميزاتا وعيوبها و تحديد الفجوات التي تحتاج لمعالجة.
- 3- تقديم إطار عمل متكامل لتخمين جهود فريق العمل بالاعتماد على معايير متنوعة بعد تحليل الأطر المتوافرة.

1 . 4- فرضيات البحث:

باعتبار كون تكلفة الجهد للمشاريع هي الأكثر جدلاً في الآونة الأخيرة، نهدف لأن نقدم في هذا المشروع نموذج عمل جديد لتقدير الجهد اعتماداً على الدراسة التحليلية للمشروع، لذا من المهم جداً أن تكون الدراسة التحليلية مكتوبة بدقة وبخبرة لتمكنا من إعطاء النتيجة الأدق للجهد المطلوب لتنفيذ المشروع وتحديد عدد الساعات الكلي وعدد الأشخاص المطلوب توافرهم لإنجاز المشروع خلال فترة معينة.

1 . 5- النتائج المتوقعة والجهات المستفيدة:

- بناء إطار عمل متكامل يساعد مدراء المشاريع والمطورين في التقدير الأمثل لجهود فريق العمل في تنفيذ مشاريع الويب بأفضل كفاءة وجودة ممكنة.
- حل المشكلات التي تعاني منها شركات البرمجة السورية الصغيرة منها والكبيرة من أجل الاستفادة القصوى من مهارات فريق العمل في مشروع الويب.

2- الجزء النظري

يمكن تصنيف التخمين البرمجي إلى ثلاث مراحل:

- المرحلة الأولى: تشمل تخمين الحجم
- المرحلة الثانية: تتمثل في تخمين الوقت وتخمين الجهد
- المرحلة الثالثة: تتمثل في تخمين الكادر وتخمين الكلفة.



شكل رقم 1 تسلسل مراحل التخمين في دورة حياة تطوير البرمجيات

2 . 1- المنهجيات العامة لتقدير الجهد:

هي عديدة كما ذكرها Boehm عام 1981 حيث منها ما هو نماذج حسابية Models Algorithmic ومنها غير حسابية Non-Algorithmic Models ويمكن تلخيصها بالتالي:

2. 1. 1 - رأي الخبير (Expert Judgment) :

هنا يتم الاستفادة من الخبراء المتخصصين في مجال تطوير المشاريع البرمجية حيث يقوم كل منهم بإعطاء تقدير تكلفة المشروع ومن ثم يجري مناقشة التقديرات والوصول إلى رأي نهائي متفق عليه لهذه التكاليف.

2. 1. 2- التقدير المبني على التشابه (Estimation by Analogy):

هنا يتم تقدير تكلفة المشروع بناء على تكلفة مشاريع مماثلة سابقة ، حيث يتم تقدير تكلفة المشروع قيد الدراسة بالتشابه مع ما سبقه، وتمتاز هذه الطريقة بالدقة إذا توافرت بيانات لمشاريع مماثلة للمشروع الذي هو قيد التقدير، أما ما يعيبها أنها مستحيلة الحدوث إذا لم يتواجد هناك مشاريع مماثلة سابقة.

2. 1 . 3- قانون باركنسون (Parkinson's Law) :

هو قانون قائم على فكرة أن العمل يتوسع كي يكون على اتساع الوقت المتاح لإنجازه، وهذا يعني أنه عند تحديد وقت أكبر لإنجاز عمل ما، فإن هذا العمل يميل إلى استغراق الوقت المحدد له، فإذا تم تحديد ثلاث ساعات على سبيل المثال لمجموعة من الأشخاص لتنفيذ مهمة محددة مقابل أربعة ساعات لمجموعة أخرى من الأشخاص لتنفيذ المهمة نفسها، فإنّ كلتا المجموعتين تنتهيان المهمة في حدود الوقت المحدد لها، وهذا يعني أن تقدير التكاليف يتم بناء على المدة المحددة لتنفيذ المشروع والموارد المتاحة، مثال على ذلك: إذا طلب تسليم مشروع برمجي خلال 12 شهراً ولدينا 5 مبرمجين فإنّ الجهد اللازم للإنجاز يتم تقديره بـ 60 شخص*شهر.

2. 1 . 4- التسعير للفوز (Pricing to Win):

هنا يتم تقدير تكلفة المشروع البرمجي بناءً على مقدرة العميل على الدفع عليه، أي أنّ الجهد يقدر بناء على ميزانية العميل وليس على أداء البرمجية، يتم اللجوء إليها في حالة عدم توافر مواصفات تفصيلية للمشروع المطلوب تنفيذه، وتقدر التكاليف بحيث تتوافق مع أساسيات العرض المقدم بدون النظر إلى جودة أداء النظام المستهدف.

2. 1 . 5- تقدير التكلفة (Cost Estimation) :

يتطلب تخطيط المشروع البرمجي بشكل عام تقديم تقديرات بخصوص المعالم الأساسية للمشروع مثل الحجم، الموارد، فريق التطوير، الجداول الزمنية، والتكلفة. ويتحكم بتطوير المشاريع البرمجية أربعة عوامل أساسية هي: الوقت، والمتطلبات، والمخاطر، والموارد التي تشتمل على فريق العمل، العامل المادي، والتجهيزات والبنية التحتية، وأي تغيير غير متوقع في هذه العوامل سوف يؤثر مباشرة على خطة تطوير المشروع.

هنا يجب التأكيد أنه في الغالب التكاليف في معظم المشاريع هي تكلفة الجهد، حيث أن تكلفة العتاد اللازمة تعد نسبياً بسيطة من ناحية التقدير، أما تكاليف الجهد لا تشمل فقط رواتب وأجور فريق تطوير المشروع البرمجي بل تحمّل الجهات المطورة التكاليف الأخرى ذات الصلة بنسب معينة على الرواتب والأجور بما يضمن الوصول إلى تكلفة العمل الكلية.

2. 1 . 1 - 6 عدد أسطر الكود البرمجي (SLOC) Source line of code :

هو مقياس يستخدم لتقدير حجم برنامج عن طريق تعداد الأسطر البرمجية في نص البرنامج. يستخدم SLOC للتنبؤ بمقدار الجهد الذي سيكون مطلوباً لتطوير البرنامج، وكذلك لتقدير إنتاجية البرمجة أو الصيانة بمجرد أن يتم إنتاج البرنامج. وهناك نوعان رئيسيان لقياس ال SLOC: قياس فيزيائي (مادي) وقياس منطقي، حيث المقياس المادي هو عدد أسطر الكود في البرنامج بعد إهمال أسطر التعليقات، أما المقياس المنطقي فيُعرّف بأنه عدد التعليمات القابلة للتنفيذ ويرتبط ذلك بلغة البرمجة فمثلاً في لغة البرمجة باسكال يمكن تعريفه بأنه عدد الفواصل المنقوطة التي تنتهي بها التعليمات.

2. 1 . 1 - 7 نموذج بناء التكلفة (COCOMO) Constructive Cost Model: [2]

تم تطوير نموذج بناء التكلفة من قبل Boehm عام 1981 وهو من النماذج التي تُستخدم لتقدير الجهد اللازم لإنتاج نظام برمجي محدد وبالتالي التكلفة والوقت الزمني له محددين. عملية الحساب في هذا النموذج تتم باستخدام معادلة الجهد وذلك لتقدير حجم العمل المطلوب مقاساً بالواحدة شخص*شهر وبقية نتائج هذا النموذج مثل تقدير المتطلبات والصيانة تشتق من نتيجة هذه المعادلة. وافترض Boehm أنه يمكن تصنيف المشاريع البرمجية حسب تعقيدها وحجمها إلى ثلاث فئات :

أ- المشاريع العضوية organic : وهي المشاريع البرمجية البسيطة والصغيرة نسبياً، وتعمل فيها فرق برمجية صغيرة العدد ذات خبرة جيدة بتطوير التطبيقات، وتكون متطلبات المشروع مفهومة والمتطلبات أقل صرامة .

ب- المشاريع القليلة الترابط semi-detached : وهي المشاريع البرمجية المتوسطة التعقيد والحجم ، وتعمل فيها فرق برمجية لديهم مستويات خبرة متفاوتة، وتكون متطلبات المشروع هنا متنوعة بين صارمة وأقل صرامة.

ج- المشاريع المضمنة embedded : هي المشاريع التي لها خصوصية معينة، وهي خاضعة لمجموعة قيود عملية وبرمجية خاصة.

وقد تم اقتراح نماذج مختلفة من Cocomo وذلك بناء على مقدار الدقة المطلوبة وهم:

أ- النموذج الأساسي Basic Cocomo

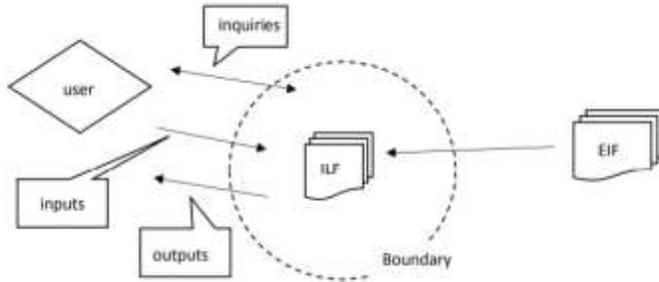
ب- النموذج المتوسط Intermediate Cocomo

ج- النموذج المفصل Detailed Cocomo

2. 1 . 8 - تحليل النقاط الوظيفية Function point analysis [3]:

وهو كما عرفه Albrecht مقياس كمّي يستخدم للتعبير عن كمية المهام التي يجب أن يحققها نظام المعلومات المستهدف حيث يُنظر إلى النقطة الوظيفية كوحدة قياس لتقدير حجم كل وظيفة على حدى في المشروع (FSM) Function Size Measurement، ويعتمد حساب تكلفة كل وحدة استناداً إلى مشاريع سابقة. يتم تحديد المتطلبات الوظيفية الخاصة بالمنتج البرمجي، ومن ثم يتم تصنيف كل مطلب ضمن واحد من خمسة أنواع:

مخرجات - مدخلات - ملفات داخلية - واجهات خارجية - استعلامات، والشكل الآتي يوضح ما سبق:



شكل رقم 2 أنواع المتطلبات الوظيفية

حيث يجب في البداية تحديد النطاق الذي يفصل بين المشروع الذي يجب قياسه وبين التطبيقات الخارجية والمستخدمين، ثم يتم تقييم تعقيدات هذه المتطلبات وتحديد عدد النقاط الوظيفية غير المعدلة (UFP) Unadjusted Function Point.

2. 1 . 9 - نقاط حالة الاستخدام Use Case Point (UCP): [4]

تم تطوير هذه الطريقة من قبل Karner عام 1993 وهي تقنية تُستخدم لتوقع حجم البرمجيات المستخدمة لإنهاء تطوير مشروع برمجي، وهي تشبه ال FPA من حيث المبادئ لكنها مُوجهة لتقدير حجوم المشاريع البرمجية غرضية التوجه Projects

Unified Object Oriented، وهي تعمل حصراً عندما يتم استخدام منهجية Unified Modeling Language (UML) و Rational Unified Process (RUP) عند تصميم البرمجيات وتطويرها، وذلك بسبب أن UCP تعتمد في حساب حجم المنتج البرمجي على متطلبات النظام التي تتم نمذجتها باستخدام حالات الاستخدام Use Case والتي هي جزء من تقنيات ال UML ، فضلاً عن أن هذه التقنية تأخذ بالاعتبار عند تقدير الجهد اللازم للمشروع العوامل التقنية و بيئة العمل . وتعتمد هذه التقنية في حساب حجم المشروع البرمجي على عملية توازن بين أوزان العناصر التالية:

- عدد حالات الاستخدام في النظام وتعقيدها.
- عدد الفاعلين في النظام وتعقيدهم.
- متطلبات غير وظيفية.
- البيئة التي سيتم تطوير المشروع فيها.

3- الدراسات المرجعية

لقد قَدّم الباحثون في هذا المجال عدة دراسات بما يخص تقدير الجهود، ومنها الدراسات التالية:

دراسة أعدها (Suresh Nageswaran، 2001) بعنوان Test Effort Estimation Using Use Case Points^[5] قدمت هذه الدراسة تجارب على تقدير الجهود باستخدام نموذج Use Case Points واستخلصت أنه هناك مزايا عملية كبيرة من استخدام هذا النموذج، ولكنه بحاجة للمزيد من البحث والتجريب للحصول على المزيد من الفوائد الجوهرية في الوصول إلى طريقة موضوعية للتحقق من صحة التقديرات.

دراسة أعدها (Taghi M. Khoshgoftar، Ali Idri ، Alain Abran، 2002) بعنوان Estimating software project effort by analogy based on linguistic values^[6]: في هذه الدراسة قام الباحثون بتصميم أسلوب جديد بالاعتماد

على المنطق الضبابي والمحددات (quantifiers and fuzzy logic) كذلك على الاستدلال اللغوي باستخدام أسلوب المقارنة (linguistic reasoning by analogy) لتخمين جهد المشروع البرمجي وأطلقوا على هذا الأسلوب اسم المقارنة الضبابية (Fuzzy Analogy). ووفقاً لهذا البحث تم تطبيق النموذج المقترح على COCOMO dataset إذ لوحظ أن الطريقة المقترحة قد حسنت عملية التخمين و حصلت على نتائج أفضل من الطرائق التقليدية وأكثر دقة.

دراسة أعدها (Dharmender Singh Kushwaha, Ashish Sharma, 2012) بعنوان Estimation of Software Development Effort from Requirements Based Complexity [7] هدفت هذه الدراسة إلى اقتراح مقياس لتقدير جهد تطوير المشاريع بناءً على تعقيد متطلبات المشروع التي سيتم تطويرها والتي بدورها تعتمد على توصيف متطلبات النظام للمشروع المقترح (software specification requirements)، وللتحقق من صحة النتائج تمت مقارنتها مع تقديرات بطرق أخرى (algorithmic use case point, function point count, models, lines of code)، وتم في هذه الدراسة إنشاء نموذج يستند إلى وضع علامات أو تصنيفات لمتطلبات النظام المطلوبة وقد ثبت أن تعقيد متطلبات البرنامج له تأثير مباشر على الجهد اللازم وبناءً على ذلك اقترح Kushwaha & Sharma معيار تحسين تقدير المتطلبات على أساس تعقيدها وذلك بناءً على الـ SRS المُعطى وتبين أن التقديرات التي تعتمد على هذه المنهجية أكثر فائدة لأنه يتم معرفة تعقيد المتطلبات و بالتالي تقدير الجهد بمرحلة مبكرة جداً.

دراسة أعدها (Abdul Wahid, Mudasir Manzoor Kirmani, 2015) بعنوان Revised Use Case Point (Re-UCP) Model for Software Effort Estimation [8] أكدت هذه الدراسة أن الجانب الذي لا غنى عنه في أي شركة تطوير للبرمجيات هو تشكيل آلية للتعامل مع المشكلات التي تؤدي إلى الفوضى. يتمثل أحد المجالات للتعامل مع هذه المشكلات في جدولة عملية التطوير بأكملها للخضوع لعملية

تقدير مناسبة وفعالة ، حيث يمكن تقدير جميع الموارد مسبقًا بشكل جيد من أجل التحقق مما إذا كان المشروع المتصور مجدياً وضمن الموارد المتاحة. اللبنة الأساسية في أي تصميم موجه للكائنات هي مخططات حالة الاستخدام التي يتم إعدادها في المراحل الأولى من التصميم بعد فهم المتطلبات بوضوح. تعتبر مخططات حالة الاستخدام مفيدة لتقريب تقديرات مشروع تطوير البرمجيات. يعطي هذا العمل البحثي نظرة عامة مفصلة عن طريقة Re-UCP (Revised Use Case Point) (نقطة حالة الاستخدام المنقحة) لتقدير الجهد لمشروعات البرمجيات. طريقة Re-UCP هي طريقة مُعدلة تعتمد على طريقة UCP لتقدير الجهد. في هذه الدراسة البحثية ، خضع 14 مشروعًا لتقدير الجهود باستخدام طريقة Re-UCP وتمت مقارنة النتائج مع نماذج UCP و e-UCP. وتظهر المقارنة بين الـ 14 مشروعًا أن Re-UCP تفوقت بشكل ملحوظ على تقنيات UCP الحالية و (e-UCP) Effort Estimation Techniques.

دراسة أعدها (Alessio Petrozziell, Federica Sarro ، 2018) بعنوان Linear Programming as a Baseline for Software Effort Estimation: [9] قدمت هذه الدراسة طريقة جديدة تعتمد البرمجة الخطية وتم تسميتها ب Linear Programming for Effort Estimation (LPEF) وقد تم مقارنة هذه الطريقة بنموذج سابق المستخدم الوحيد في البرمجة الخطية وهو (Automatically Transformed Linear Model (ATLM) وبينت الدراسة أن LP4EE أكثر دقة من ATLM بنسبة 17% من التجارب ، وذلك من حيث تقسيم البيانات المختلفة في 44% من الحالات.

دراسة أعدها (Chastine , Daniel O. Siahaan, Rahmi Rizkiana Putri) بعنوان Improve the Accuracy of Software Project Effort and Cost Estimates in COCOMO II Using GWO: [10] في هذه الدراسة يتم توضيح أنّ قيمة دقة COCOMO II أقل دقة لأنه لا يزال هناك فرق كبير بين جهد المشروع الفعلي وقيمة التكلفة المقدرة. ولتحسين دقة COCOMO II ، يتم

استخدام طريقة (GWO) Gray Wolf Optimization التي تعتمد على سلوك الذئب في اصطیاد الفرائس. في هذه الدراسة ، يتم استخدام COCOMO II GWO للحصول على مستوى أعلى وأكثر دقة من دقة التقدير وتقليل قيمة الخطأ الإجمالية أو متوسط الخطأ النسبي (MMRE) لمشاريع البرمجيات. إن نتيجة الاختبار MMRE التي أنتجتها الدراسات السابقة لهذه الدراسة التي تستخدم COCOMO II BCO (تحسين مستعمرة النحل) كان 12.92%. وفي الوقت نفسه ، فإن معدل MMRE بالطريقة المقترحة COCOMO II GWO هو 1.731%. هذا يعني أن الطريقة المقترحة يمكن أن تقلل من قيمة الخطأ في MMRE بنسبة 11.19%.

دراسة أعدها (Preeti ، Verma Aditya ، 2021) بعنوان Calibrating Intermediate COCOMO Model Using Genetic Algorithm: [11] هدفت هذه الدراسة إلى تعديل معاملات نموذج COCOMO الوسيط باستخدام خوارزمية الكشف عن مجريات الأمور. ستم مقارنة التكلفة المقدرة بالبيانات الحقيقية. يتم تطبيق المنهجية على مجموعة بيانات برنامج COCOMO NASA ، وتم استخلاص أنه لا بُد من استخدام تقنية البرمجة الجينية (Genetic Programming) لبناء هيكل نموذج مناسب لتقدير جهد البرنامج.

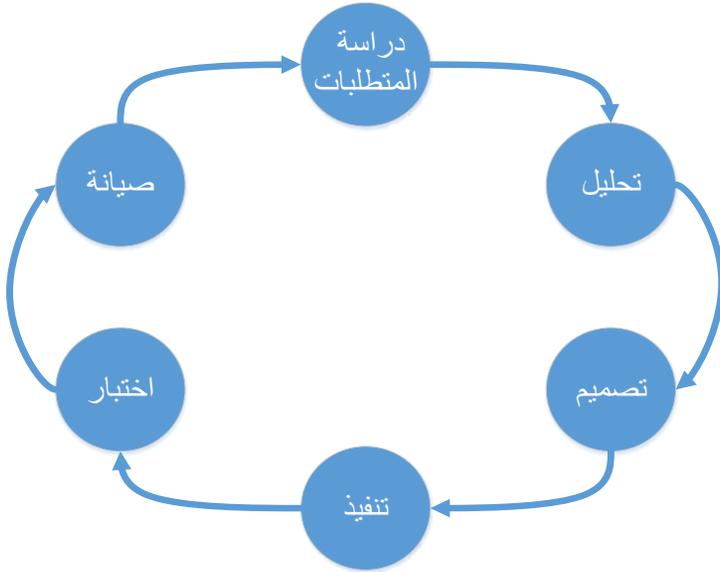
بالرغم من جميع الدراسات ولكن لم يتم التوصل إلى قيم دقيقة وقريبة من الواقع في هذا المجال، وأيضاً لم يتم التوصل إلى نسب توفير كبيرة يمكن اعتمادها لتوفير الوقت المخصص لانجاز المشاريع وبالتالي التكلفة الكلية للمشاريع.

4- الحلول المقدمه:

بعد التمعن في الدراسة المرجعية السابقة ودراسة المنهجيات المتبعة لتقدير الجهد تبين أنه من الممكن الاستفادة من عدة عوامل لإنتاج أسلوب عمل مميز لتقدير الجهد البرمجي المبذول لأي مشروع بأكثر قيمة تقريبية للعمل الحقيقي.

تم ملاحظة أن تقدير الجهد يجب أن يكون على مستوى المشروع بشكل كامل، وليس فقط على مستوى مرحلة واحدة وهي البرمجة، بل أن التقدير الصحيح للمشروع يجب أن يكون شامل لجميع المراحل التي يمر بها منذ البداية حتى النهاية، وهي [12]:

- دراسة المتطلبات
- تحليل المشروع
- تصميم المشروع
- التنفيذ والتحقق البرمجي للمشروع
- اختبار المشروع
- نشر وصيانة المشروع



شكل رقم 7 مراحل انجاز المشروع البرمجي

وبصرف النظر عن المنهجية المتبعة لتطوير المشروع إن كانت شلالية أم حلزونية، حيث أن الحلزونية هي تكرار لخطوات العمل منذ البداية ومن الممكن في بعض الحالات أن تحصل على نتيجة المنهجية الشلالية مضروبة بعدد دورات المشروع.

وفي النهاية سيتم اعتماد فكريتي النقاط والتثقييل، بالنسبة للنقاط فيتم حسابها من كل عامل على حدى، وبعد ذلك يتم تثقييل هذه النقاط ليحصل المشروع على نتيجة معينة لنتمكن من مقارنتها مع مشاريع أخرى لمعرفة الفروقات بين الجهود المبذولة على كل مشروع. يتطلب تخطيط المشروع البرمجي بشكل عام تقديم تقديرات بخصوص المعالم الأساسية للمشروع مثل الحجم، الموارد، فريق التطوير، الجداول الزمنية، والتكلفة. ستكون الآلية البرمجية تعتمد على مجموعة من الأسئلة تشمل جميع حيثيات المشروع على جميع مراحل الستة، وستكون مجموعة كبيرة جداً من الأسئلة وذلك لضمان الحصول على أكبر قدر ممكن من الدقة في تقدير التكلفة للمشروع البرمجي. سيتم الاعتماد على مفهوم الكتل البرمجية المطلوب إنجازها بوضع مجموعة متدرجة من الأصناف ووضع تثقييل لكل صنف كتلة برمجية مطلوب. وتكون الأصناف كالتالي:

- 1- صنف بسيط: يكون لعمليات الإدخال البسيطة، مثل تسجيل معلومات طالب، أو إنشاء حساب مستخدم (الاسم - العنوان - الهاتف - الايميل ... الخ).
- 2- صنف علمي (يتطلب إنجاز معادلات رياضية).
- 3- صنف معقد (خوارزميات).
- 4- صنف دلالي (عمليات دلالية أو شبكات عصبونية بدرجات تعقيد متغيرة).
- 5- صنف معالجة (يتطلب معالجة بيانات موزعة أو متزامنة ... الخ).
- 6- صنف مرئي (معالجة صور).

مع إمكانية تحديد حجم هذا الصنف (صغير - متوسط - كبير). سيتم الاعتماد على البيانات المقدمة من دراسة المتطلبات الخاصة بالمشروع بالإضافة للدراسة التحليلية، أي أنه سيتم تنفيذ حسابات تقدير الجهد بعد تلك المرحلتين، بحساب الجهد المُقدم للمرحلتين السابقتين (دراسة متطلبات - دراسة تحليلية)، بالإضافة لتقدير جهد المراحل التالية (تصميم وتنفيذ واختبار ونشر وصيانة). إحدى أهم النقاط الواجب أخذها بعين الاعتبار هو وجود الكيان المادي الممكن أن يكون جزء من العمل ويتطلب مهام إضافية (تجميع - إعداد - برمجة خاصة).

أما بالنسبة لعدد الأسطر البرمجية، فلا يمكن الوصول إليه إلا عند انتهاء العمل البرمجي، لذا لا يمكن الاعتماد عليه في تقدير تكلفة العمل البرمجي عند بدايته، بينما من الممكن الاعتماد عليه لتقدير جهد مشروع تم إنشاؤه مسبقاً ليتم مقارنة مشروع آخر معه، لذا تم استثناءه من فكرة المشروع.

احتساب نتيجة تقدير الجهود Effort Estimation Result Calculate:

في نهاية تقييم الجهود وحساب التتقيات يتم تحديد الجهد بالنسبة للوقت وعدد الأشخاص بإعطاء عدد من الساعات بناءً على عدد الأشخاص. وسيكون الناتج عبارة عن عدد طبيعي أي موجب، وبدون فواصل وذلك للسماح بإجراء عمليات رياضية على الناتج بشكل بسيط وواضح.

وبعد ذلك يتم تحديد إما الوقت أو عدد الأشخاص ليتم تحديد المعامل الآخر بشكل تلقائي. حيث أنه عند زيادة الوقت يقل عدد الأشخاص، بينما في حال إنقاص الوقت يتم زيادة عدد الأشخاص المطلوبين لإنجاز العمل. كما يتم تحديد عدد الأيام اعتماداً على نوع العمل الموكل إلى العاملين إن كان بوقت جزئي Part Time أو بوقت كامل Full Time، وبذلك يتم إعطاء التقدير النهائي حسب نوع العمل وعدد الأشخاص.

مثال توضيحي:

ليكن لدينا مشروع تم تقدير جهده ب 800 ساعة عمل، عند تقسيم هذا الناتج على عدد الأشخاص يعطي عدد الأيام.

تقدير الجهد اعتماداً على عدد الأشخاص:

عند تحديد عدد الأشخاص سينتج معنا عدد الأيام. فرضاً 10 أشخاص سيقومون بهذا العمل، أي أن العمل يستغرق 80 ساعة للإنجاز، وهذا لا يعني أن نقوم بتقسيم 80 على 24 لمعرفة عدد الأيام اللازمة لإنجاز العمل، أي أن نتيجة القسمة وهي 3 أيام و8 ساعات تعتبر نتيجة خاطئة للعمل، بالرغم من كونها القيمة الفعلية المقدره لإنجاز العمل.

ولتقدير العدد الكلي للأيام سيتم توزيع هذه الساعات حسب نوع العمل إن كان Part Time أو Full Time، حيث أن العمل بوقت جزئي Part Time يعتبر 4 ساعات يومياً، بينما العمل الكامل Full Time هو 8 ساعات يومياً.

على هذا الأساس فإن العمل المقدر بـ 800 ساعة عمل وموزع على 10 أشخاص 80 ساعة عمل كوقت إجمالي لكل شخص، يكون تقديره بالأيام بالنسبة للعمل بوقت جزئي Part Time هو حاصل قسمته على 4 أي 20 يوم، بينما يكون تقديره بالأيام بالنسبة للعمل بوقت كامل Full Time هو حاصل قسمته على 8 أي 10 أيام.

تقدير الجهد اعتماداً على عدد الأيام:

عند تحديد عدد الأيام سينتج معنا عدد الأشخاص.

على سبيل المثال تم تحديد عدد الأيام بـ 25 يوم، فيكون عدد الأشخاص الواجب تعيينهم بوقت جزئي للعمل Part Time هو حاصل قسمة 800 على 25 وهو عدد الأيام وبعدها القسمة على 4 وهو عدد الساعات في اليوم كالتالي:

$800 \div 25 = 32 = 4 \div 8$ أي أنه يكفي تعيين ثمانية أشخاص لتنفيذ المشروع خلال 25 يوم. أما عدد الأشخاص الواجب تعيينهم بوقت كامل للعمل Full Time هو حاصل قسمة 800 على 25 وهو عدد الأيام وبعدها القسمة على 8 وهو عدد الساعات في اليوم كالتالي: $800 \div 25 = 32 = 8 \div 4$ أي أن أربع أشخاص كافيين للقيام بالعمل خلال 25 يوم .

5- التنفيذ والاختبار:

بهدف الحصول على قيم واقعية، قمنا أولاً بالاستعانة بخبراء (في عدة مننديات برمجية والغالبية لم يذكروا أسماؤهم) لتحديد قيم التنقيلات وحصلنا على آراء متقاربة، ولتحديد القيم بشكل أدق قمنا بإنشاء استبيان الكتروني للوصول إلى القيم الصحيحة لتثقيل كل قيمة يتم اختيارها. وبعدها قمنا بمعالجة النتائج واختيار قيم التثقيل اعتماداً على الاختيار الأكثر انتقاءً أو قيمة تثقيل متوسطة بين قيمتين متقاربتين الانتقاء. وتم تخزين جميع القيم

ضمن قاعدة المعطيات، للتعامل معها برمجياً مع إمكانية تغيير القيم دون العودة إلى الشيفرات البرمجية.

بعد تحليل نتائج الاستبيان وجدنا من خلال الإجابات أن:

أ- المعايير الأساسية الواجب تقدير الجهود حسبها هي: نوع المشروع، حجم المشروع، العتاد الصلب.

ب- في حال تم تقسيم حجم المشروع إلى 5 مستويات (صغير جداً، صغير، متوسط، كبير، كبير جداً) فإن الزمن المُقدّر لكل مستوى: صغير جداً (خلال مدة أقصاها أسبوع)، صغير (خلال مدة أقصاها شهر)، متوسط (خلال مدة أقصاها 3 أشهر)، كبير (خلال مدة أقصاها سنة)، كبير جداً (خلال مدة أقصاها 5 سنوات).

ج- في حال برمجة تطبيق Desktop: بالنسبة لكل نموذج يتم إنشاؤه هناك وقت مخصص لتصميم النموذج، وبتقسيم حجم العمل التصميمي لكل عنصر إلى خمسة مستويات للتصميم (بسيط جداً - بسيط - متوسط - متقدم - متقدم جداً) فإن الوقت المُقدّر لكل مستوى بالدقائق: بسيط جداً: دقيقة، بسيط: دقيقتان ونصف، متوسط: خمس دقائق، متقدم: سبع دقائق ونصف، متقدم جداً: عشر دقائق.

د- فيما يخص التعديل الفوتوغرافي على الصور: في حال تم التقسيم إلى ثلاث مستويات للتصميم (بسيط - متوسط - متقدم) فإن الوقت المُقدّر لكل مستوى بالدقائق: بسيط: 10 دقائق، متوسط: 50 دقيقة، متقدم: 100 دقيقة.

هـ- في حال برمجة موقع ويب: فإن الوقت المُتوقع لتصميم صفحة: تم اعتماد النتيجة: ساعة.

ح- في حال برمجة موقع ويب: بالنسبة لكل صفحة يتم إنشاؤها هناك وقت مخصص لتصميم الصفحة، وبتقسيم حجم العمل التصميمي إلى خمسة مستويات للتصميم (بسيط جداً - بسيط - متوسط - متقدم - متقدم جداً) فإن الوقت المُقدّر لكل مستوى بالدقائق: بسيط جداً: دقيقة، بسيط: دقيقتان ونصف، متوسط: خمس دقائق، متقدم: سبع دقائق ونصف، متقدم جداً: عشر دقائق.

ط- في حال برمجة موقع ويب : لتحسين مظهر الإظهار قمنا بافتراض ثلاث تقنيات، (CSS -Bootstrap - كل عنصر له تنسيقه اليدوي الخاص) فإنّ الوقت المُقدّر لكل تقنية بالدقائق: Bootstrap : ربع ساعة، CSS: نصف ساعة، تنسيق يدوي: ساعة.

ي- الوقت المُقدّر لإضافة جدول والوقت المُقدّر لإضافة حقل إلى هذا الجدول مع تحديد القيود الخاصة به: (القيم بالثواني): تم اعتماد الناتج 15 ثانية.

ك- بعد إنشاء جدول وتحديد حقوله، ووجود بيانات بحاجة للإدخال ضمنه. فإنّ الوقت المُقدّر لإضافة قيمة ضمن خلية في جدول ما (بالثواني): 10 ثواني.

ل- بعد إنشاء جدول وتحديد حقوله، فإنّ الوقت المقدر لإنشاء مفتاح أجنبي (بالثواني): تم اعتماد القيمة 60 ثانية.

م- الوقت المتوسط المُقدّر لإنشاء عملية (كالإجراءات والوظائف والقوالب) متضمنة وقت التفكير بالعملية (القيم بالدقائق): بسيط: 5 دقائق، متوسط: ربع ساعة، متقدم: ساعة.

ن- الوقت المُخصّص لكتابة الكود الخاص بالكتلة البرمجية اعتماداً على تعقيد العملية مُوزّعة على المستويات التالية (بسيطة جداً - بسيطة - متوسطة - معقدة - معقدة جداً) : بسيطة جداً : ربع ساعة، بسيطة : ساعة، متوسطة: ساعتين ونصف، معقدة: خمس ساعات، معقدة جداً : عشر ساعات.

س- الوقت المخصص لكتابة الكود الخاص بالكتلة البرمجية اعتماداً على صنف العملية مُوزّعة على المستويات التالية (بسيط - علمي - معقد - دلالي - معالجة - مرئي): بسيط: ربع ساعة، علمي: ساعة، معقد: ساعتين ونصف، دلالي: خمس ساعات، معالجة: عشر ساعات، مرئي: خمسة عشر ساعة.

ع- فيما يخص العتاد الصلب: فإنّ الوقت المُقدّر لإنجاز أبسط مشروع باستخدام التقنيات التالية: آردوينو: ساعتان, Pic16: 5 ساعات, متحكم آخر: 10 ساعات, متحكمات تحتاج إلى تصميم دائرة الكترونية: 25 ساعة, PLC: 30 ساعة.

ف- أيضاً فيما يخص العتاد الصلب : حسب حجم المشروع، فإنّ معامل المضاعفة الذي سيؤثر على إعداد العتاد الصلب: صغير جداً: 1, صغير: 2, متوسط: 5, كبير: 10, كبير جداً: 25.

ص- الوقت المُقدّر للصيانة والاختبار والنشر, تم اعتماد القيم التالية: زمن الصيانة 10% من زمن المشروع, زمن الاختبار 10% من زمن المشروع, زمن نشر التطبيق ساعة واحدة, زمن نشر الموقع 4 ساعات.

ولتنفيذ العمل فُمنّا بتطوير موقع ويب يحتوي على التبويبات التالية:

- 1- إضافة مشروع جديد Add New Project: لإضافة مشروع برمجي واحتساب الجهد اللازم لتنفيذه.
- 2- عرض المشاريع View Projects : لعرض قائمة بالمشاريع التي تمت إضافتها مسبقاً، مع إمكانية عرض تفاصيلها.
- 3- التثقيلات Widths: لعرض التثقيلات التي تم تخزينها في قاعدة المعطيات، مع إمكانية تغييرها.
- 4- COCOMO: لاحتساب تثقيل المشروع اعتماداً على نموذج COCOMO
- 5- Use Case Point: لاحتساب تثقيل المشروع اعتماداً على نموذج Use Case Point

حيث يتم أولاً إضافة مشروع جديد ووضع تفاصيل المشروع بناءً على الدراسة التحليلية والتي تم حصرها بالبنود التالية:

1- اختيار بيانات رئيسية عن المشروع (اسم المشروع - نوع المشروع - حجم المشروع - العتاد الصلب المستخدم)

2- إن كان نوع المشروع تطبيق، يقوم بإدخال تصميم النماذج الموجودة فيه، كل نموذج على حدى وتحديد البيانات الخاصة بكل نموذج والتي هي (اسم النموذج - مستوى التصميم - عدد العناصر - مستوى التعديل الفوتوغرافي - عدد الصور التي بحاجة إلى تعديل فوتوغرافي).

3- إن كان نوع المشروع موقع، يقوم أولاً بإدخال بيانات التصميم الخارجي للصفحات بدون العناصر، كل صفحة على حدى والتي هي: (هل الصفحة MasterPage - عدد الصفحات - مستوى التصميم - أسلوب التنسيق المعتمد لكل صفحة)، ومن ثم يقوم بإدخال بيانات تصميم الواجهات التي يمكن أن يتم وضعها في صفحة واحدة أو عدة صفحات في حال الحاجة والبيانات هي (اسم الواجهة - مستوى التصميم - عدد العناصر - مستوى التعديل الفوتوغرافي للصور - عدد الصور التي بحاجة إلى تعديل فوتوغرافي).

4- ادخال بيانات قاعدة المعطيات في حال وجودها، والتي هي (عدد الجداول - عدد حقول كل جدول - عدد الجداول المترابطة - عدد الجداول التي بحاجة إلى ادخال بيانات مسبق - حجم البيانات المدخل - عدد العمليات المستخدمة (الاجراءات والقوادح) - حجم العمليات (بسيط - متوسط - متقدم).

5- ادخال بيانات الكتل البرمجية كل كتلة على حدى، وهي (اسم الكتلة - وصف الكتلة - تعقيد العملية - نوع العمليات ضمنها حسب الأصناف).

يتم تخزين قيم تثقيلات البنود التي تم اختيارها في قاعدة المعطيات ومن ثم إجراء عمليات حسابية لاحتساب الناتج النهائي لتثقيل الجهود، ومن ثم تحويل قيمة التثقيل إلى ناتج بالدقائق والساعات، مع إمكانية إضافة احتساب مراحل الاختبار والصيانة والنشر إلى الناتج النهائي، ليصبح بالإمكان احتساب تقدير الجهد بعد توزيعه على الطاقم البرمجي واحتساب عدد المبرمجين والأيام اللازمة لإنجاز المشروع. تكمن المرحلة الثانية

بحساب تثقيل المشروع باستخدام خوارزميتي (COCOMO و Use Case Point). أما المرحلة الثالثة والأخيرة، فهي تحويل تثقيل المشروع الناتج عن النموذجين السابقين إلى ساعات ثم إدخاله إلى واجهة المقارنة، ليتم عرض مقارنة بين هذين الناتجين، والناتج الذي توصلنا إليه ليتم عرض مخطط يبين الفروقات بين النواتج الثلاث.

6- النتائج:

تم تجريب العديد من المشاريع البرمجية موثقة بدراسة تحليلية، وتم اعتماد مثال مشروع برمجي وهو موقع خاص بدار للأيتام¹، عن طريق إدخال البيانات اعتماداً على الدراسة التحليلية ضمن واجهات موقع الويب الذي قمنا بتطويره. وكانت النتيجة كالتالي:

تثقيل المشروع بشكل كامل: 32405
الوقت النهائي المقدر بالدقائق: 19443
الوقت النهائي المقدر بالساعات: 324:3

شكل رقم 8 نتيجة تقدير الجهد للمشروع

وعند توزيع العمل على طاقم العمل فإنه عندما يعمل الطاقم عمل جزئي ستكون النتيجة كالتالي:

¹ <https://github.com/rawannsr/orphans>

تجهيز العتاد الصلب: لا يوجد
التصميم: يحتاج إلى 1 شخص/أشخاص خلال 11 يوم
قاعدة المعطيات: يحتاج إلى 1 شخص/أشخاص خلال 5 يوم
التحقيق البرمجي: يحتاج إلى 1 شخص/أشخاص خلال 50 يوم
اختبار المشروع: يحتاج إلى 1 شخص/أشخاص خلال 7 يوم
صيانة المشروع: يحتاج إلى 1 شخص/أشخاص خلال 7 يوم
نشر المشروع: يحتاج إلى 1 شخص خلال 1 يوم
المشروع بشكل كامل: يحتاج إلى 5 شخص/أشخاص خلال 75 يوم

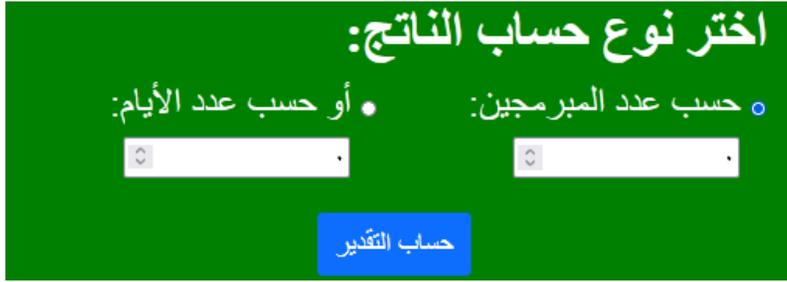
شكل رقم 9 توزيع العمل على طاقم يعمل جزئي

أما عندما يعمل الطاقم عمل كامل تكون النتيجة كالتالي:

تجهيز العتاد الصلب: لا يوجد
التصميم: يحتاج إلى 1 شخص/أشخاص خلال 5 يوم
قاعدة المعطيات: يحتاج إلى 1 شخص/أشخاص خلال 3 يوم
التحقيق البرمجي: يحتاج إلى 1 شخص/أشخاص خلال 25 يوم
اختبار المشروع: يحتاج إلى 1 شخص/أشخاص خلال 3 يوم
صيانة المشروع: يحتاج إلى 1 شخص/أشخاص خلال 3 يوم
نشر المشروع: يحتاج إلى 1 شخص خلال 1 يوم
المشروع بشكل كامل: يحتاج إلى 5 شخص/أشخاص خلال 37 يوم

شكل رقم 10 توزيع العمل على طاقم يعمل كامل

كما من الممكن تحديد عدد المبرمجين أو عدد الأيام واحتساب ناتج تقدير الجهد اعتماداً على ذلك، كالتالي:



شكل رقم 11 حساب التقدير

فرضاً تم تحديد عدد المبرمجين ب 5 مبرمجين، ستكون النتيجة كالتالي:



شكل رقم 12 حساب التقدير حسب عدد المبرمجين

وفي حال تحديد عدد الأيام ب 10 أيام، ستكون النتيجة كالتالي:

شكل رقم 13 حساب التقدير حسب عدد الأيام

من الضروري بعد ذلك الذهاب إلى تبويب COCOMO و Use Case Point لاحتساب تقدير الجهد حسب النموذجين ووضع النواتج ضمن واجهة المقارنة كالتالي:



شكل رقم 14 مقارنة النتائج

تم تنفيذ التجريب على مشروع آخر خاص بتوثيق أجهزة الصرافة²، وكانت النتائج كالتالي:

Effort Estimation: 39, COCOMO: 200, Use Case Point:370

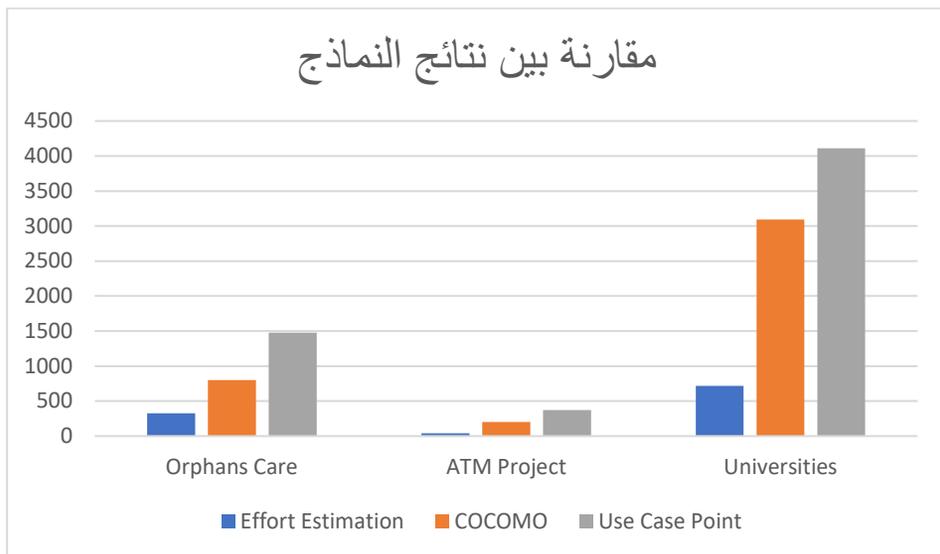
وأيضاً على مشروع دلالي للوصول إلى معلومات حول الجامعات³، وكانت النتائج كالتالي:

² https://github.com/rawannsr/ATM_Project

³ <https://github.com/rawannsr/Universities>

Effort Estimation: 715, COCOMO: 3091, Use Case Point: 4112

ليتم إظهار مخطط يبين الفروقات بين النواتج كالتالي:



شكل رقم 15 مخطط توضيحي لمقارنة النتائج

وكما هو ملاحظ، فقد توصلنا لفارق كبير بتحديد الجهد اللازم لإنجاز المشروع عن النموذجين المشهورين COCOMO، و Use Case Point. حيث تم تحقيق نسبة توفير بمقدار 60% عن نموذج COCOMO، ونسبة توفير بمقدار 78% عن نموذج Use Case Point بالنسبة لمشروع Orphans Care، أما بالنسبة لمشروع الصرافة، فتم توفير 81% عن نموذج COCOMO (حيث نتجت لدينا النسب وفق التالي: بتقسيم ناتج برنامجنا على ناتج النموذج فينتج لدينا نسبة هي نسبة إنجاز المشروع المطبق مقارنة بنتيجة النموذج الثاني وفق التالي: $0.195 = 200 \div 39$ وبالتالي نسبة التحسن تُحسب وفق $1 - 0.195 = 0.805$ وحتى نأخذها كنسبة مئوية نضرب بـ 100 فينتج 80.5% أي تقريبا 81%)، ونسبة توفير بمقدار 90% عن نموذج Use Case Point، (حيث طريقة حساب النسب السابقة هي نفسها كما ذكر في الأعلى)، ونسبة توفير بمقدار 90% عن نموذج Use Case Point. أما بالنسبة لمشروع الجامعات، فتم توفير 77% عن نموذج COCOMO، ونسبة توفير بمقدار 83% عن نموذج

Use Case Point. حيث أن هذا التوفير هو توفير الوقت المخصص لإنجاز المشروع.

7- الخاتمة والتوصيات:

قمنا في هذا البحث بالتوصل إلى خوارزمية تعمل على تخمين جهود فريق العمل للمشاريع البرمجية وذلك بعد إجراء استبيان فيه مجموعة كبيرة من الأسئلة تشمل جميع حيثيات المشروع من دراسة المتطلبات، تحليل، تصميم المشروع، التنفيذ، التحقيق البرمجي، الاختبار، نشر وصيانة المشروع، وتم عرضه على عدد كبير من الخبراء في مجال التطوير البرمجي، ثم استنتجنا بواسطته عدد كبير جداً من العوامل التي تؤثر على تقدير الجهود بالإضافة إلى تنقيحات هذه العوامل ثم بناء موقع برمجي للشركات البرمجية يحسب تقدير الجهود البرمجية لأي مشروع ويب وفق عدد من الأسئلة الدقيقة التي يعرضها الموقع عما يحتاجه المشروع من متطلبات دقيقة ثم يعرض لنا الجهد المتوقع لفريق العمل مع إمكانية تحديد عدد المبرمجين لإنجاز العمل عندها يعرض عدد الأيام التي يحتاجونها أو تحديد عدد الأيام التي يحتاجها المشروع لينتهي خلاله عندها يعرض عدد المبرمجين الذين نحتاجهم لإنجازه مع الأخذ بعين الاعتبار الدوام إن كان جزئي أم كامل، ثم قارنًا هذه الطريقة بمنهجين سابقين في هذا المجال وذلك بإدخال مجموعة كبيرة من المشاريع البرمجية وتم ذكر مشروعين برمجيين كمثالين على خوارزمتنا وعلى المنهجين السابقين وتوصلنا بواسطة خوارزمتنا إلى نتائج مُحسنة. يُنصح بتوفير خدمة حساب الجهد بما يخص الشيفرات المكتوبة بحساب عدد الأسطر البرمجية باعتبار البنى الشرطية و الحلقات التكرارية والاجرائيات المستخدمة، وذلك لكي يتم تحديد عدد الأسطر التي سيتم تنفيذها بأقرب تقدير ممكن. كما يُنصح بإضافة قدرات لفريق العمل، حيث أن المبرمج الخبير من شأنه أن ينجز العمل المؤكل إلى مبرمج متوسط بفارق كبير في الوقت، لذا من الجيد اعتبار الخبرة البرمجية لكل فرد من فريق العمل.

المراجع

1. JOWHRY, D., 2018 - **Effort Estimation of software development**. The Higher Institute for Applied Sciences and Technology.
2. BOEHM, B., 1981 - **Software Engineering Economics**. New York, 197.
3. RASK, R., LAAMANEN, P., & LYYTINEN, K., 1992 - **A Comparison of Albrecht's Function Point and Symons' Mark II Metrics**.
4. SAROHA, M., & SAHU, S., 2015 - **Tools & Methods for Software Effort Estimation Using Use Case Points Model—A Review**. In International Conference on Computing, Communication & Automation, pp. 874-879.
5. NAGESWARAN, S., 2001 - **Test Effort Estimation Using Use Case Points**. In Quality Week, Vol. 6, pp. 1-6.
6. IDRI, A., ABRAN, A., & KHOSHGOFTAAR, T. M., 2002 - **Estimating Software Project Effort by Analogy Based on Linguistic Values**. In Proceedings Eighth IEEE Symposium on Software Metrics, pp. 21-30.
7. SHARMA, A., & KUSHWAHA, D. S., 2012 - **Estimation of Software Development Effort from Requirements Based Complexity**. Procedia Technology, 4, 716-722.
8. KIRMANI, M. M., & WAHID, A., 2015 - **Revised Use Case Point (Re-UCP) Model for Software Effort Estimation**. International Journal of Advanced Computer Science and Applications, 6(3), 65-71.
9. PETROZZIELL, F. S., 2018 - **Linear Programming as a Baseline for Software Effort Estimation**. ACM Transactions on Software Engineering and Methodology.
10. PUTRI, R. R., SIAHAAN, D. O., & FATICHAH, C., 2021 - **Improve the Accuracy of Software Project Effort and Cost Estimates in COCOMO II Using GWO**. In ICICoS, pp. 128-133.
11. VERMA, A., & PREETI., 2021 - **Calibrating Intermediate COCOMO Model using Genetic Algorithm**. IEEE.
12. RIBDAWI, G., 2018 - **Software Engineering**. Syrian Virtual University.