Budapesti Műszaki és Gazdaságtudományi Egyetem

Természettudományi Kar

Gyártási folyamatok statisztikai minőségirányítása megerősítéses tanulással

TDK Dolgozat

Készítette:

Pálvölgyi Bence Tamás

Konzulens:

Dr. Viharos Zsolt János

Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI)
Mérnöki és Üzleti Intelligencia Kutatólaboratórium
Intelligens Folyamatok Kutatócsoport

2021

Tartalomjegyzék

Kivonat

A gyártási folyamat határozza meg az alapanyag és a végtermék közti út árát. Mivel a gyárak között óriási és nagyon éles gazdasági verseny áll fent, ezeket a folyamatokat a lehető leghatékonyabban akarják elvégezni. Erre nagyon sok módszert használnak, az egyik, hogy tömegtermelés esetén, a folyamatosan keletkező gyártási adatokat statisztikailag elemzik. Ilyen téren a mesterséges intelligencia jól alkalmazható, hiszen adatok közti összefüggéseket kell feltárni és kiaknázni.

Eddig a tanulási módszerek közül főleg a felügyelt tanulás (supervised learning) és a nem felügyelt tanulás (unsupervised learning) trejedt el, mi egy másik, a megerősítéses tanulást (reinforcement learning) választottuk. Erről meglehetősen kevés cikk van gyártási területen, viszont egy nagyon fontos előnye az előbbiekhez képest, hogy gyorsan képes alkalmazkodni a környezeti változásokra. A széles körben használt Q Table módszert alkalmaztuk a kiszámíthatóbb és könnyen áttekinthető eredmények eléréséhez, ezen felül implementáltuk néhány új megerősítéses tanulás bővítményt, mint például az ön-szabályzó felfedezés-kiaknázás arányt, hatékony frissítési szabályt és az általunk bevezetett Reusing Window-t és a Measurement Window-t.

Az eddigiekben a környezet pontos szimulációjával tettük lehetővé, hogy a tanuló ágens ezen kísérletezzen és használja fel a fontos akciókat. Az algoritmus fő célja, hogy a lehető legkevesebb gyártási költséggel a lehető legtöbb sikeres végterméket hozza létre, költségoptimális módon.

A Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI) kezdeményezésével, a részben a Mesterséges Intelligencia Nemzeti Laboratórium keretein belül, az Opel Szentgotthárd Kft.-vel együttműködve valós helyzetben kutathattuk ezt a témát. Egy adott termék gyártási folyamatából egy résznek a megmunkálását vizsgáltuk múltbéli és jelenlegi adatok alapján.

A kutatás közeljövőbeli célja az, hogy ezt a tudást valódi gyártósorba beépítve is kiaknázzuk.

Abstract

The manufacturing process determines the value difference between the raw material and the end product. The factories are in a big race in term of economy and this brings the focus on the optimality of this process. There have been many solutions, one of those is to analyze the production data. Artificial Intelligence is perfect for such use, as you have to find coherence between these values.

The three main type of machine learning algorithms are supervised learning, unsupervised learning and reinforcement learning, which we have chosen to use in our research. There are many articles on this field with supervised and unsupervised learning but not so much about reinforcement learning, which has the main advantage of being able to adapt to sudden environmental changes. Our reinforcement learning algorithm works with the widely used Q table method, because of its simplicity in implementation and clarity. We also upgraded this learning agent with functions from state-of-the-art articles, such as the self-regulating epsilon values of exploration-exploitation, effective update rule, and the developed by us Reusing Window and Measurement Window.

Until this time, we used a simulation of the environment for the agent to be able to learn and use the important values and reactions. The main goal of the algorithm is to produce the most successful product while also keeping the cost to the lowest level.

With the initiation of the Számítástechnikai és Automatizálási Kutatóintézet (SZTAKI), within the framework of the Artificial Intelligence National Laboratory and with the collaboration with Opel Szentgotthárd Kft. we could explore and test our learning agent on data from a real world production line. They provided the data from past and ongoing manufacturing processes of a single product, from which our environment simulation was made.

Future goals include the implementation of the program into a real production line.

1. Tanulási módszerek és alkalmazásuk a gyártásban

A Mesterséges Intelligencia (MI, Artificial Intelligence, AI) és a Gépi Tanulás (Machine Learning, ML) már az élet minden területén jelen van, ez alól a gyártási szektor se kivétel. Ezen a területen a fejlődés mértéke egyre csak nő, sok az új felfedezés, és az ezekről szóló cikkek száma is bővül, amelyek tanulmányozása még sok kutatómunkával és teszteléssel jár. Az ezen területen kívüli MI és Gépi Tanulási fejlesztéseket is figyelni kell, hogy párhuzamosan tudjuk ezeket beépíteni a kódba. A globális Ipar 4.0 program is ezt ösztönözi: ” az eszközök önállóan kommunikálnak egymással az értéklánc mentén: a jövő egy olyan „okos” gyárát hozva létre ezzel, amelyben a számítógép-vezérelt rendszerek nyomon követik a fizikai folyamatokat, létrehozzák a fizikai valóság virtuális mását és decentralizált döntéseket hoznak önszervező mechanizmusok alapján.” [61].

A Mesterséges Intelligenciának nagyon sok ága van: gépi tanulás, kereső algoritmusok, többszempontú optimalizálás, következtetési és szakértői rendszerek, gráf modellezések és bejárások, …; ezek közül a Mély Tanulás (Deep Learning) kapja jelenleg a legnagyobb figyelmet. A Mesterséges Intelligencia legfőbb építőeleme a Gépi Tanulás, ennek feltalálásától is hívjuk ezeket MI-nek. Eleinte csak két féle tanulási mód volt, a felügyelt tanulás (supervised learning) és a nem felügyelt tanulás (unsupervised learning), de a nyolcvanas években Richard S. Sutton és munkatársai megalkották a megerősítéses tanulást (Reinforcement Learning, RL). Azóta ezeknek az ágaknak vannak kombinált verziói is, például a félig felügyelt tanulás (semi-supervised learning). Az irodalomkutatásból az látszik, hogy míg a felügyelt és a nem felügyelt tanulás sűrűn használt statisztikai minőségirányítás téren, a megerősítéses tanulás szinte kizárólagosan csak a gyártás ütemezés és a robotika területein terjedt el.

1.1. Megerősítéses Tanulás a Gyártásban

A megerősítéses tanulás gyártási ütemezésre való felhasználása egyre elterjedtebb, változatos tanuló algoritmusokkal, mint például a Q-tanulás [2][3][4], mély Q-tanulás [5] és adaptált verziói a Q-tanulásnak [6][7]. A legtöbbjük érték-alapú algoritmus [2][3][4][5][8], de

van néhány politika-alapú is [4][7]. Sokan „epszilon-kapzsi” metódust, míg W.Bouzza és munkatársai [2] emellett még saját gépi választási szabályt is használtak a felfedezésre. Vannak kutatások [2][5][10], ahol több ágenses tanulást alkalmaztak egy helyett. Ezeket főleg szimulációs környezetben tesztelték. Például Kardos Cs. és munkatársai Q-tanuláson alapuló megerősítéses tanulást használtak ütemezési/szétosztási döntések megoldására termelési rendszerben és szimuláció segítségével bebizonyították, hogy az ő megoldásuk lényegesen csökkenti a gyártási megrendelésekhez tartozó átfutási időt [9]. Ezen felül az is látszott, hogy a gyártási környezet bonyolultságának növekedésével még hasznosabbá válik a megerősítéses tanulás (a felügyelt és nem felügyelttel szemben).

Robotikai felhasználásánál A. Nair és munkatársai bemutatottak egy megerősítéses tanulást tartalmazó rendszert, ami képes volt megoldani többlépcsős feladatokat [11]. Plappert M. és munkatársai [12] ellenőrzési feladatok sorozatát vetették fel és konkrét kutatási ötleteket hoztak fel a megerősítéses tanulás fejlesztésére. Y. Zhu és munkatársai kombinálták a megerősítéses és az imitációs tanulást, hogy megoldjanak manipulációs feladatokat pixelek alapján [13]. G. Kahn és munkatársai egy kiválóan teljesítő megerősítéses tanuló algoritmust írtak egy robot irányításához [14]. P.Long optimalizált egy decentralizált érzékelő szintű ütközés elkerülés politikát megerősítéses tanulással [15], T. Johannink és munkatársai kutatták a hagyományos visszacsatolásvezérlési módszereket [16]. Van néhány próbálkozás megerősítéses tanulást alkalmazó felhasználásokra gyártási folyamatirányításban, de azok speciális esetekre vonatkoznak és sokkal ritkábbak, mint az ütemezési és robotikai cikkek, ezért ezeket részletesebben is meg kell vizsgálni.

Az egyik speaciális eset feladata a sóoldat injektálása szalonnás élelmiszeripari termékekbe, amelyhez egy adaptív vezérlési modell szükséges. A befecskendezési nyomást és a befecskendezési időt egy megerősítéses tanulású, mély determinisztikus politikai gradienssel (deep deterministic policy gradient) lehet állítani, amit R.E. Andersen és munkatársai [43] mutattak be. A mély determinisztikus politikai gradiens képes volt adaptálni egy modellt egy adott szimulált környezethez, ami 64 valós kísérleten alapult. A 15%-os tömegnövekedést megcélozva, végül 14.9%-os tömegnövekedést értek el 2.6%-os szórással.

G.Beruvides és munkatársai [44] legfőbb hozzájárulása egy megerősítéses tanulás architektúra megtervezése és implementálása volt, alapértelmezett mintázat azonosító többlépcsős összeszerelési folyamatokban, fémlemez alkatrészekkel, ahol a mintatesztelést

szimuláción végzik. Ez az architektúra három módból állt össze (tudás, kognitív és végrehajtás módok), kombinálva egy mesterséges intelligencia modell könyvtárát egy Q tanuló algoritmussal. A három módszerrel (MLP, SOM és MLP + GA) bemutatott eredmények nagy pontosságot értek el a speciálisan ehhez a képzéshez és validációhoz generált különböző mérési paraméterek tekintetében. Ez az algoritmus bővítmény megerősítéses tanulással (ebben az esetben Q tanulással) további előnyökhöz is vezetett, mert segít paramétereket meghatározni a modelleknek, amik pontosabb illesztéseket tesznek lehetővé a tesztelés során.

F. Guo és munkatársai [45] bevezetett egy profi, hibrid keretrendszert egy üveglencse fröccsöntési folyamatának optimalizálásra és irányítására. Ez a modell szimulációval előre betanított, aztán ezt a tudást felhasználva optimalizálja és irányítja a gyártósort. Ez egy kiváló cikk, az itt használt megerősítéses tanuló algoritmus a többi genetikus algoritmusot és az fuzzy következtető algoritmusot lényegesen felülmúlta optimalizáció és szabályzás szempontjából.

N. Khader és S.W. Yoon [3] áramköri lapok felületszerelési folyamatának paraméterbecslésére szolgáló módszert hoztak létre. A céljuk az volt, hogy fokozzák a forrasztópaszta átvitelét a folyamat során egy optimális adaptív szabályzóval. Nyomtatási sebesség, nyomtatási nyomás és az elválasztás sebessége alkotta az állapotterét a megerősítéses tanulásnak, és két predikciós modell állapítja meg a jutalom függvényt, ami az átlaga és a szórása a forrasztópaszta tértfogat átviteli hatékonyságának. A közvetlen jutalom, az átlag és szórás kiszámolása után a C_{pk} és C_{pkm} értéket meghatározza minden komponensre, ebben a példában ez a módszer harmonizál a statisztikai minőségirányítással. Ha az elöbb kiszámolt értékek egy határérték fölött vannak, akkor maga a jutalon 1, különben -1. A szimuláció alapú tesztelés azt mutatta, hogy a tanuló ágens sikeresen eljutott a végső állapotba viszonylag kevés akciót felhasználva.

F. Li egy új neurális hálózatot [46] javasolt, amelyet Reforcement Learning Unit Matching Recurrent Neural Network-nek (RLUMRNN) hívnak, azzal a céllal, hogy megoldja azt a problémát, hogy a tipikus neurális hálózatok általánosítási teljesítménye és nemlineáris közelítési képessége nem szabályozható a rejtett rétegszám és a rejtett réteg csomópontszámának tapasztalatalapú kiválasztása miatt.

Ezt a rejtett szint számának és a rejtett csomópontok számának tapasztalat alapú kiválasztása okozza. Felhasználói oldalon egy ”megkülönböztetőt” vezettek be, hogy a gördülőcsapágyak állapotát három monoton trend osztályba sorolják be: emelkedő trend,

ereszkedő trend és statikus trend. A megerősítéses tanulás tulajdonságából kifolyólag a visszacsatolt neurális hálókban lévő rejtett szintek és rejtett csomópontok száma monoton trend egységekbe lettek besorolva. Ennek a koncepciónak az előnyeiből egy új állapot trend becslő metódust hoztak létre. Ebben a becslő metódusban a szinguláris spektrális entrópia mozgó átlagát használták az állapot megkülönböztető funkciójára, és aztán ez a funkció lett beírva a modellbe, hogy megbecsüljék a gördülőcsapágyak állapot trendjét. Ez az elgondolás lett lemásolta és tovább fejlesztetve R. Wang és munkatársai [47] által, azzal a különbséggel, hogy hatékonyabb mély tanuló hálózati felépítés szerepelt benne, és a tesztelő környezet egy próba-berendezés és egy mozdonycsapágy alkotta.

P. Ramanathan és munkatársai [48] létrehoztak egy megerősítéses tanuláson alapuló okos szabályzót, aminek a teljesítménye egy nem lineáris kúpos tartályrendszer vízszintjének irányításán lett demonstrálva. Az előnye, hogy ez egy egyedülálló szabályzó, előre beprogramozott környezet vagy tudás nélkül. Hardweres megvalósításával bebizonyosodott, hogy a nem lineáris kúpos tartályrendszer folyadékszintjét hatékonyan állította és kompenzálta a véletlen zavarokat a rendszerben. Ez a szabályzó hatékonyabb a PID, fuzzy és egyéb neurális háló alapú megoldásoknál, mert elkerüli a nem lineáris funkciók lineárissá alakítását, PID paraméterek hangolását, átviteli függvények használatát, és fuzzy tagsági függvények kifejlesztését. Egy lényeges előnye ennek az ötletnek az, hogy a megerősítéses tanulás minden alkotórésze valós alkalmazáshoz van kötve, és valós adatokkal van visszacsatolva, ez emeli ki ezt a kutatást és az eredményeit. *A valóságbeli használatban az ágens csak kiaknázó döntéseket hozott, míg a szimulációban különböző felfedezés-kiaknázás arányok voltak*. Először a kezdeti Q mátrix egy szimulációból keletkezett, amit MATLAB-ban futtatak, valós összeköttetés nélkül. Ez a Q mátrix lett később frissítva valós kísérleti adatok kiaknázásával.

Sok eset áttekintése után még egy fontos szempont kerül előtérbe: a megerősítéses tanulás tanuló lépéseinek száma, ezzel együtt a szimuláció is, amiben a lépések megtörténnek. A gyártási ütemezés egy számítógépen fut, ami összeköttetésben áll a gyártó rendszerrel, így természetes, ha ”előrejátszuk” a jövőbeli eseteket (tervezés, gyártási lépések stb), részben azért mert megvan a hozzá szükséges információ. Hasonló a helyzet a robotikai felhasználásoknál is, ahol a háromdimenziós mozgástér lehetséges lépéseit meg tudjuk vizsgálni. Ebből kifolyólag mind a két terület használ szimulációt, ami megkönnyíti a megerősítéses tanulást. *Az összes eddig felsorolt kutatásban szerepelt szimuláció a környezetről*, ami vagy egy fizikális modellről

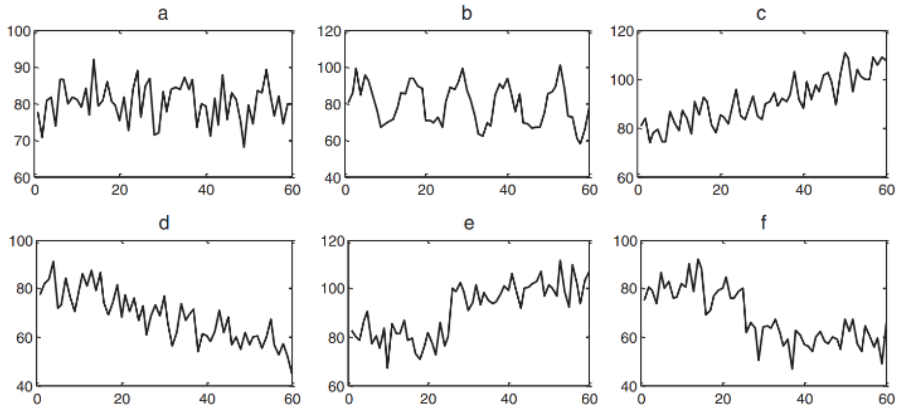
készült, vagy valós adatokra épült. *A szimulációs megközelítés szinte kötelező a megerősítéses tanulásban, mert sok tanuló lépésre van szükség*, ami a valóságban időigényes, kockázatos vagy nagyon drága lenne. A mi kutatásunk is szimuláción dolgozik, amíg a statisztikai minőségirányítás valós adatokra épül. Megtörtént események adatait és szakemberek segítségét használtuk a pontos szimuláció felépítése érdekében.

A megerősítéses tanulás felhasználása különböző ipari/gyártási területen még sok nyitott kérdést és feladatot ad, ez igaz a statisztikai minőségirányításra is. Az előbb bemutatott speciális esetekhez képest a mostani kutatás egy általánosabb, szélesebb körben alkalmazható megerősítéses tanulást használó statisztikai minőségirányítást mutat be. Az ágens fő célja, hogy a gyártást a felső tolerancia limit és az alsó tolerancia limit között tartsa a statisztikai szabályzó diagrammon, költségoptimális módon, de ezen kívül több új kiegészítést is kidolgoztunk.

1.2. Statisztikai Minőségirányítás a Gyártásban

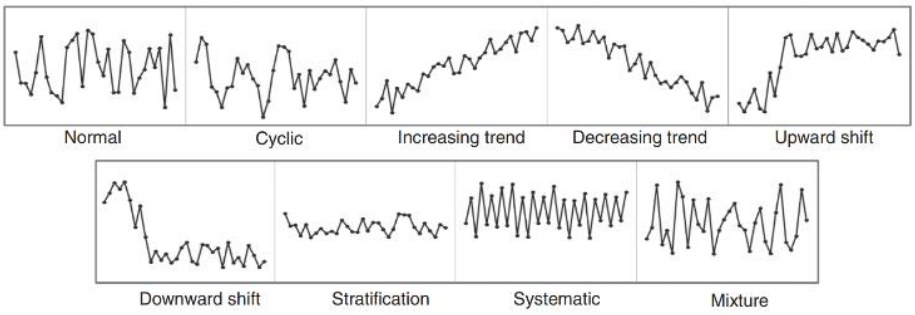
A következőekben a legfrissebb elemzéseket és kulcs-példákat mutatok be, a Mesterséges Intelligencia statisztikai minőséggyártásban történő alkalmazásokhoz.

A gyártási statisztikai minőségirányítást (statistical process control) a tudományos irodalom szabályzó diagram mintaként (control chart pattern) említi, azaz az szabályzó diagramnak és a trendviselkedésnek kulcsfontosságú szerepe van a minőségirányításban. A következőekben kulcs-példákat mutatok be, a Mesterséges Intelligencia statisztikai minőségirányításban történő, legfrissebb alkalmazásához. V. Ranaee és A. Ebrahimzadeh [19] hat típusba osztja be a tipikusan előforduló trendeket, amit bemutat az 1.2.1 ábra.



1.2.1 ábra

Egy másik beosztás K. Lavangnananda és S. Khamchai [20] alapján kilenc különböző mintát emel ki (1.2.2 ábra), amelyből az utolsó különböző effektek kombinációjaként áll elő, következésképp az egyes minták egyszerre is előfordulhatnak.



1.2.2 ábra

Az alábbi beosztások alapján még mindig kérdése, hogy mit értünk ”normális” viselkedés alatt, milyen eloszlást, milyen paramétereket. Egy probléma megállapítani, hogy az szabályzó diagramon éppen milyen zaj van, annak milyen eloszlása van még akkor is, ha a normál trend és a zaj elkülöníthető [21]. Ugyanakkor egy valós szabályzó diagrammon lévő különböző trendek definiálásához tartozó paraméterek meghatározása is további feladat. A diagrammok számos változatát használják a gyártás iráynítására, a főbb példákat mutatja be a következő fejezet.

G. Köksal és munkatársai számos adatbányászatú technika minőségirányítási körben való felhasználását vizsgálták [22]. Négy fő csoportot különítettek el: termék/folyamat minőségi leírás, minőség előrejelzés, minőségi osztályozás és paraméter optimalizálás. Bebizonyították az efféle kutatások fontosságát és felhasználhatóságát az iparban.

T.T. El-Midany és munkatársai [23] mesterséges neurális hálókat használtak, hogy felismerjék a többváltozós abnormális minták alosztályait egy kompresszor egyik fő alkatrészének, a forgattyúház megmunkálásában. Valós és szimulált adatokon is dolgoztak, és azonosítani tudták az abnormális mintákat okozó változókat.

V. Ranaee A. Ebrahimzadeh hibrid intelligens módszert használtak [19], hogy eldöntsék, hogy egy folyamat a tervezett módon fut vagy természetellenes mintái vannak. Ez a módszer három modulból állt: egy jellemző generáló modulból, egy több osztályú Support Vector Machine (SVM)-on alapuló osztályzó modulból (MCSVM) és egy genetikus algoritmust

használó optimalizáló modulból. Az algoritmusokat szintetikusan generált szabályzó diagramokon tesztelték.

K. Lavangnananda és S. Khamchai [20] különböző zajú mintákat elemeztek. Ők három osztályzót használtak: döntési fát, mesterséges neurális hálót és önbeállító társulási szabály generátort (Self-adjusting Association Rules Generator), szabályzó diagram mintákhoz, amiket GARH modellel (Generalized Autoregressive Conditional Heteroskedasticity), előre meghatározott egyenletekkel generáltak $\bar{X}$ diagramhoz.

G.D. Pelegrina és munkatársai [24] többféle vakforrás elkülönítő (Blind Source Separation) módszert használtak párhuzamos vezérlőtáblák keverésének feloldására, hogy magas minősítési arányt érjenek el.

H. De la T. Gutierrez és D.T. Pham [25] bemutatott egy új rendszert a gépi tanulás képzési mintáinak létrehozására: Support Vector Machine (SVM) és Probabilistic Neural Network (PNN).

W.-A. Yang és munkatársai [26] egy hibrid megközelítést használtak, az extrém-pont szimmetrikus módú dekompozíciót, extrém tanulógéppel (extreme learning machine), hogy meghatározzák az egyidejű szabályzó diagram mintákat.

A.R. Motorcu és A. Güllü X-R [27] szabályzó diagramot hozott létre egy gyár minden egyes gyártósorára, hogy kiváló minőségű termelést biztosítson fontos problémák megoldásával: nemkívánatos tűréshatárok, rossz felületkezelés vagy kerekség gömb alakú öntöttvas alkatrészek a megmunkálása során.

T. Huybrechts és munkatársai [28] szabványosítotást, trendmodellezést és autoregresszív mozgóátlag (autoregressive moving average) modellt alkalmaztak, hogy meghatározzák a rövid távú korrelációt a rákövetkező mérések között. A kontrollon kívüli megfigyelések a Dijkstra-modellel pontosan meghatározhatók a mért és előrejelzett értékek közötti korrigált reziduumok kumulatív összegző diagramján. Az esettanulmányhoz két automata fejőrendszerrel és egy hagyományos fejőrendszerrel működő gazdaság tejhozam-adatait használták fel.

Viharos Zs. J. és Monostori L. [29] már 1997-ben bemutatott egy megközelítést a folyamatláncok mesterséges neurális hálózatokkal és genetikai algoritmusokkal történő optimalizálására minőségellenőrzési diagramok segítségével. Kimutatták, hogy a „belső” paraméterek (időbeli paraméterek a gyártási lánc mentén) ellenőrzése szükséges, így korai döntéseket lehet hozni, hogy egy adott alkatrész gyártását folytatni kell-e vagy sem, ezzel a

megoldással lehetséges a gyártási rendszer folyamatos optimalizálása. Viharos Zs. J. és Kis K. B. [30] a neuro-fuzzy rendszerekről és azok műszaki diagnosztikában és mérésben való alkalmazásáról készített felmérést, további gépi tanulási technikák megnövelték a forgácsolókerámiák marószerszám-élettartamát a rezgésjelek megfelelő tulajdonságainak kiválasztásában [31].

Az alkalmazott technikákat tekintve a legelterjedtebb megközelítések statisztikai módszereken alapulnak, mint például autoregresszió, mozgóátlag és ezek kombinációi: autoregresszív integrált mozgóátlag modell (autoregressive integrated moving average model) [32] lineáris regresszióanalízis, kvázi-lineáris autoregresszív modell [33] vagy Markov-láncmodell (Markov chain models) [34]. Ezek a módszerek múltbéli gyártási vagy aktuális adatokon alapulnak a modellezéshez és az előrejelzéshez.

Egy másik megközelítés jelent meg a mesterséges intelligencia fejlődésével, mint például a mesterséges neurális hálók (artificial neural networks), tartó vektorgépek (support vector machines) vagy a mintasorozat hasonlóságán alapuló legközelebbi szomszédos megközelítésekkel való modellezések [35]. Ezen a területen számos görbeillesztési módszer létezik kis mintaadatokhoz, például genetikai algoritmus [36]. A mesterséges neurális hálózatok statisztikai módszerekkel kombinált alkalmazása az egyes megközelítések hátrányainak kompenzálására jobb osztályozási és közelítési eredményeket eredményez a trend előrejelzésben.

R. Paggi és munkatársai [37] egy vegyes, valós folyamatméréseket integráló fizikai modellt mutattak be a folyamatok prognózisos értékein túlmutató bizonytalanságok kiszámítására. Az ismert függőségeket különféle fizikai modellezési technikák, mint a végeselemes módszerek, analitikai egyenletek reprezentálhatják. Francesco és munkatársai [38] a hasonlósági tesztekből származó hatékony méréseket használtak a hátralévő hasznos élettartam (remaining useful life) értékelésének pontosságának javítására.

A megfelelő statisztikai minőségirányítást szabályzó diagramok alkalmazásának fontosságát Wu és munkatársai [49] eredményei tükrözik. A globális navigációs műholdrendszer (global navigation satellite system) deformációs információinak azonosítására és riasztására új módszert javasoltak, amely kombinálja az ensemble-integrált empirikus módú dekompozíció (ensemble-integrated empirical mode decomposition) algoritmust és az EWMA vezérlőtáblát. A kísérleti eredmények azt mutatják, hogy az EWMA szabályzó diagram

módszer és a módosított EWMA módszer felismerési pontossága nagyobb, mint a kumulatív összegű szabályzó diagram módszeré. A módosított EWMA szabályzó diagram használata javítja a deformáció azonosításának és a korai figyelmeztetésnek a pontosságát, ami csökkenti a téves riasztások és a hiányzó riasztások arányát.

Aslam és munkatársai [50] egy kibővített statisztikai minőségirányítást szabályzó diagramot vezettek be, amelyet a meglévő tervvel hasonlítottak össze a neutrozófiás COM-Poisson eloszlásból származó szimulált adatok segítségével. A javasolt diagram gyakorlati megvalósítását az elektromos áramköri lapok gyártásából származó adatok felhasználásával is kifejtették. Összességében az eredmények azt mutatják, hogy a javasolt diagram hasznos kiegészítése lesz az irodalmi szabályzó diagramoknan.

Széles körben elterjedtek a vezérlőtáblák különféle alkalmazásaival kapcsolatos kutatások. V. Kavimani és munkatársai [52] alkalmazták az anyageltávolítási sebesség (material removal rate) Xbar-R diagramját és Ra irányítást használtak a Wire Electric Discharge Machining (WEDM) mérési bizonytalanságának értékelésére.

G. Aquila és munkatársai [53] a szélsebesség-tartomány államonkénti és hónaponkénti vizsgálatához az R diagramot használták. A szabályozási határértékeket a vizsgált szélenergia mezőre vonatkozó specifikus egyenlettel lehet megkapni. Az R grafikonon keresztül megfigyelhető volt, hogy a négy éven keresztül minden hónapban mért szél átlagsebességeit tekintve a nagyobb amplitúdók Bahia államban a nyári hónapokban jelentkeztek.

A Bayes-féle vezérlődiagram egy grafikus megfigyelőeszköz, amely a folyamatminták időbeli méréseit mutatja; a folyamat állapotára vonatkozó információk frissítésére a Bayes-tételt használva, ezt C.U. Mba [54] alkalmazta az újszerű hibaészlelő és -osztályozó rendszerben, amely a sztochasztikus rezonancia és a szimulált és valós sebességváltó-alkalmazásokkal tesztelt rezgésadatok rejtett Markov-modellezésén (HMM) alapul.

Az O-gyűrűk átmérőjének és vastagságának mérési hibaeredményeit G. Peng minőség-ellenőrzési táblázatán van bemutatva, amikor javítja az emberi ellenőrzésen alapuló minőség-ellenőrzést vizuális rendszerekkel [55].

A Wu és munkatársai által javasolt módszer az észlelési pontosság javítása érdekében a tervezési törvénynek megfelelően módosíthatja a vonatsínhiba-észlelési (szabályzó diagram) küszöbértékeit. Ezenkívül egy adat-előszűrő észlelés osztályozási és tanulási modellt javasoltak a homlokfelület, a hegesztési megerősítés és a csavarlyuk alsó repedésének hibajellemzőire,

amely a hibaészlelési problémát kontúrosztályozási problémává alakítja át, és az észlelési teljesítményt tovább lehet alakítani mintaképek előre betanulásával javítva [56].

P.M. Gopal és K.S. Prakash [57] minőségellenőrzési diagrammokat alkalmazott magnézium-hibrid fémmátrix kompozit előállításához, szilícium-dioxidban gazdag Ewaste CRT panelüveg és BN részecskék porkohászati úton történő megerősítésével.

K.S. Prakash és munkatársai sikeresen elkészítettek egy új Al/Rock por kompozitot keverőöntési technikával és változó erősítési paraméterek, azaz a kőzetpor részecskeméret és tömegszázalék állításával [58].

Alam és munkatársai egy veszteséges tömörítési technikát írnak le minőség-szabályzó mechanizmussal PPG megfigyelő alkalmazásokhoz [59].

A szakirodalom és a kapcsolódó alkalmazások áttekintése azt tükrözi, hogy a gyártásban többféle módszer létezik a statisztikai folyamatszabályozásra, köztük számos gépi tanulási technika is, azonban a megerősítéses tanulás vívmányait még nem használják ki ezen a termelési területen. Ez a státusz motivált minket arra, hogy az megerősítéses teanulást statisztikai minőségirányításhoz adaptáljuk, amint azt a következő bekezdésekben bemutatjuk.

1.3. Statisztikai Minőségirányítás Megerősítéses Tanulással a Gyártásban

Az általános megerősítéses tanulási megközelítésben a központi komponens egy ágens (vagy ágensek halmaza), amely érzékeli környezetét, és cselekvéseken keresztül hat a környezetére, sőt jutalmakat kap a környezettől, külsőleg és függetlenül értékelve a megtett cselekvéseket. Az ilyen interakciók sorozata folyamatos információval látja el az ügynököt, így az megszakítás nélkül tanulhat a környezetéből, sőt, mivel párhuzamosan cselekszik is, értékes feladatokat is ellát. Ez nagyon fontos előnye a megerősítéses tanulásnak a felügyelt vagy nem felügyelt technikákkal szemben, mivel a tanulási komponens folyamatosan alkalmazható az adott feladat elvégzésére, a gyártásban különösen fontos, hogy a tanulással párhuzamosan.

Leegyszerűsítetten elmondható, hogy a statisztikai minőségirányítás megerősítéses tanulással a gyártásban történő alkalmazásra javasolt jelen keretben az ágens mozgó ablakként járja végig az érzékelt idősort. Minden mozgó ablak kvantálásra kerül, és állapottá válik. Az

ágens csak a tényleges mozgó ablakot veszi figyelembe a múltból információforrásként, amikor eldönti, hogy melyik akciót válassza. A generált akciók a termelési környezetre hatnak, amellyel az előírt gyártási tolerancia tartományon belül (vagy néha sajnos azon kívül) befolyásolni tudja a trendet. Az ágens az ezzel kapcsolatos jutalmat a környezettől kapja, a megtett (termelést befolyásoló) intézkedés szerint. A javasolt külső jutalmazási rendszer úgy van definiálva, hogy minden tevékenységnek (valós) költsége van, és a jutalom fordítottan arányos a költséggel, így ösztönözve az ágenset, hogy ne válasszon drága akciót. Ezen kívül plusz büntetés jár, ha a trend (gyártott termékek) kimegy az ellenőrzési tartományból, hiszen selejtes terméket gyártott.

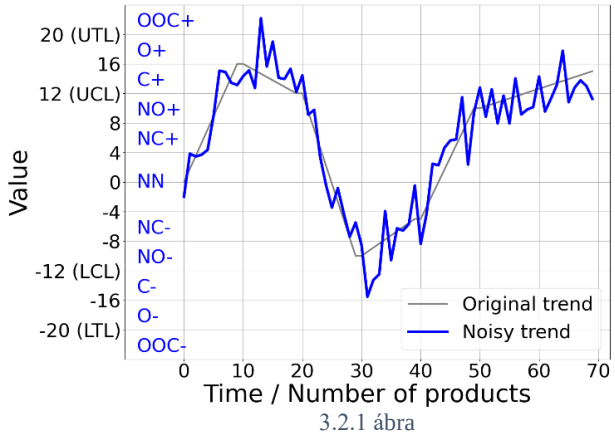
Az módszer pontosabb bemutatása érdekében a gyártási koncepcióban a megerősítéses tanulás különböző egyes összetevőit pontosan úgy kell meghatározni, ahogyan az állapotról, cselekvésekről, tanulási módszerről, jutalomról, környezeti eseményekről, tudásreprezentációról és tanulásról szóló részekben le lesz írva.

2. Gyártási folyamat bemutatása

Ahogy az előző fejezetekben már széles körben áttekintettük, a megerősítéses tanulás aktuális legkorszerűbb kutatásai és alkalmazásai a termelésben (majdnem) mindig tartalmaznak szimulációs komponenst, mint az általunk javasolt koncepcióban is. A jelenlegi megközelítésben egy szimulációs környezetet építettünk fel, amely a termelési trend viselkedését emulálja, és a gyártási környezetek és gyártóüzemek által előállítotthoz hasonló jelet generál. A környezet bármilyen hosszúságú idősoros képes szimulálni, hogy ebben az ágens tanulhasson.

2.1. Termékek, akciók, események

Az idősorok adatpontjai lépésről lépésre, eleinte zajtalanul generálódnak, vagyis egyedi lineáris mozgások/trendek alkotják őket. Az szabályzó diagramon belüli kiindulási helyzet, a lineáris vonal meredeksége és hossza szimulálja a termelési trend alakulását. A valós idősorok bonyolultsága és természetes zajossága miatt a végleges adatpont Gauss-eloszlásból lesz mintavételezve, ahol az átlaga az eredeti zajmentes (lineáris trend) pont, a zaj mértéke pedig a szórása. Számos hatékony termelési környezetben a zaj mérete a statisztikai szabályzó diagram alsó és felső tolerancia limit közti intervallum 10%-nál kisebb. Ahogy a 3.2.1. ábra mutatja, két idősor van, az egyik az eredeti trend (szürke), a másik a végső idősor, hozzáadott zajjal (kék). A tanulás csak a zajos idősorokat használja, de a környezetnek az eredeti idősorok alapján kell különféle zajszinteket generálni.



3.2.1 ábra

Mint minden termelési környezetben, bizonyos valószínűséggel előfordulhatnak független események (változások és zavarok), amelyek az idősorok alakulását is befolyásolják, a trend és a zaj állításával. A gyártás ezeket az eseményeket, mint például szerszám- vagy berendezéshiba stb., a szimulációnak emulálnia kell, emiatt meg kell határozni minden esemény gyakoriságát és hatását is. A hatások három részre oszthatók, a következőkben fogjuk ezeket részletesen megvizsgálni. Először, az események hatással vannak a trend átlagára, tehát, hogy a következő idősori pont az szabályzó diagram y tengelyén hol kezdődik. Ezeket az átlagot változtató értékeket rendszeresen eloszlásként adjuk meg, amelyek általában nem egyenletesek, tehát meg kell adni az átlagváltozások intervallumait és az egyes részintervallumok valószínűségeit. Másodszor, az események befolyásolhatják az idősor trend meredekségét, ami azt jelenti, hogy beindíthatnak például egy hosszú távú, alacsony intenzitású trendet, amely az átlagot az szabályzó diagrammon lassan lefelé vagy felfelé viszi. Harmadszor, az események befolyásolják az idősor zaját, amellyel az egyes adatpontok szórását állítják.

Ha egyszerre több esemény is bekövetkezik, hatásuk összeadódik néhány szempontból, pontosabban a következő kiindulási trend átlaga/kiinduló értéke az egyes események kiindulási pontjainak átlaga és a trend meredeksége összeadódik, a zaj azonban másként van kiválasztva: az új zaj egyenlő lesz az összes bekövetkezett esemény közüli legnagyobb zajjal.

Természetesen, ha a környezetben események fordulnak elő, amik megzavarják a trendet és a zajt, akkor azokat nekünk korrigálnunk kell, az ilyen beavatkozások az akciók. Ahogy a szimuláció generálja az idősort, a javasolt megerősítéses tanulás alapú ágens minden idősor pontban, tehát minden legyártott termék után végrehajt egy akciót, amely a környezetre is visszahat a következő pontjaiban, szintén a trend és zaj szabályzásával. Ha egyszerre több akció hajtódik végre, akkor annak hatásait az eseményekével megegyező módon számoljuk ki.

A kutatás legfrissebb verziójában már olyan valóságszerű kiegészítések vannak, mint például az úgynevezett feloldandó események. Az ilyen események minden termék legyártásánál újra bekövetkeznek, és ezzel folyamatosan megzavarják a gyártást. Az egyetlen módja, hogy ezt megállítsuk az, ha az adott eseményhez tartozó feloldó akciót végrehajtjuk. Ennek az a valóséletbeli magyarázata, hogy vannak olyan akciók, amiknek a hatásuk szinte teljesen megegyeznek, de csak egyes problémák esetén alkalmazhatók (csak erre van igazi van hatásuk).

A gyártás során valójában minden trendnek csak egy lépéshossza van, mivel az ágens minden gyártott terméknél kiválaszt és cselekszik, így minden idősor pontban a kiválasztott akció újradefiniálja a trendet és a zajszintet is, tehát a trendek ezen összetevői minden ponton folyamatosan újradefiniálódnak. Létezik azonban az akciónkak egy speciális típusa, az úgynevezett „nincs akció”, amikor az aktuális trend, zaj, stb. tényleges viselkedésében nem történik változás. Ilyenkor az eredeti trend mérete jóval hosszabb lesz egynél. Szerencsére néhány tanulási lépés után (és a valóságban is) messze a „nincs akció” a leggyakoribb, így sokkal hosszabb trendek is kialakulnak.

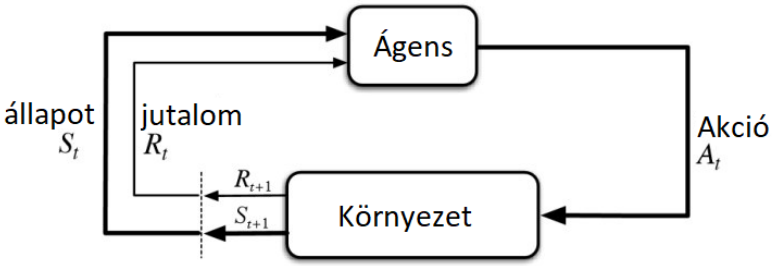
Hangsúlyozni kell, hogy adott esetben a szimulátor összes eleme egy valós gyártási környezeten alapszik, hasonlóan néhány korábban, az irodalom áttekintése során bemutatott megoldáshoz. Pontos eredményeket ad a lehetséges eseményekre, cselekvésekre, azok hatásaira, különböző zajszintekre, valamint az események gyakoriságára, cselekvési hatásaira stb., még akkor is, ha az említett ábrák torznak is tűnnek néha.

Kiemelten fontos, hogy az egyes eseményeket, akciókat, azok hatásait közösen, az ipari partnerrel együttműködésben definiáltuk, így azok tartalma a közös kutatás eredménye.

A koncepció kidolgozása és validálása során egy konkrét, valós környezetre épülő szimuláció készült. Több millió szimulált iparági adatot generáltunk, és több ezer valós értéket elemeztünk, hogy meghatározzák a valós idősorok rejtett dinamikáját a szimulációhoz. Az eseményeket valós adatok és szakértői hozzáértés segítségével meghatározott Gauss-eloszlások alapján definiáltuk.

3. A Megerősítéses Tanulás működése

A megerősítéses tanulásban többféle tanulási stratégia létezik, az időbeli különbség tanulás (Temporal Difference Learning, TD) az egyik legnépszerűbb és leghatékonyabb módszer, ezért mi is ezt a megoldást választottuk, azonban itt is alkalmazhatók más tanulási típusok is.



A TD esetben, miután az ágens megkapta a jutalmat a környezettől, a kiválasztott akció (Q) értéke az időbeli különbség egyenlettel (3.1 egyenlet) frissül, ahol Q(s,a) a Q tábla ”s” állapotában vett ”a” akció értéke.

$$Q_t(s,a) = Q_{t-1}(s,a) + \alpha\left(R(s,a) + \gamma \max_{a'} Q(s',a') - Q_{t-1}(s,a)\right)$$

3.1 egyenlet

Itt $0 \leq \alpha \leq 1$ egy állandó lépés-méret paraméter, amely befolyásolja a tanulás sebességét, ezért tanulási sebességnek nevezzük. γ egy paraméter, $0 \leq \gamma \leq 1$, ezt diszkontráta/tényezőnek nevezzük. A diszkonttényező határozza meg a jövőbeli jutalmak jelenértékét: a jövőben k időlépéssel később kapott jutalom csak $\gamma k-1$-szer kevesebbet ér, mint akkor, ha azonnal megkapná [39]. A mi esetünkben azt alkalmazzuk, hogy megbecsüljük a következő állapot értékét, ahol az ágens az ”s” állapotban végrehajtott ”a” akció után landol, R az ehhez tartozó várható jutalom. Ez a frissítési szabály egy példa egy úgynevezett időbeli különbség tanulási módszerre, mivel változásai a $Q_t(s, a)$- $Q_{t-1}(s, a)$ különbségen alapulnak, tehát két egymást követő lépés/állapot Q értékén.

Az α és γ értékeit nemrégiben az optimalizálás hiperparamétereiként kezelték, például OpenAI és Xu és munkatársai [40][41]. Számos tesztelés után azt találtuk, hogy a lépésméret paraméternek 0,3 körül, a diszkontrátának pedig 0,75 körül kell lennie az elemzett gyártási környezetben.

Az optimális akcióválasztás széles körben kutatott terület a megerősítéses tanulás területén, jól ismert „felfedezés-kiaknázás egyensúly” néven [41].

Megerősítéses tanulás esetekben, ahol nem áll rendelkezésre a probléma teljes dinamikájának modellje, szükségessé válik a környezettel való interakció, hogy „próba-szerencse” módszerrel megtanulják a cselekvés kiválasztását meghatározó optimális irányelvet. Az ágensnek fel kell fedeznie a környezetet akciók végrehajtásával és azok következményeinek érzékelésével. Egy adott időpontban egy adott politikája van, annak érdekében, hogy lássa, vannak-e lehetséges fejlesztések ezen az irányelven, az ágensnek néha különféle, esetenként nem optimális akciókat kell kipróbálnia, hogy lássa az eredményeket. Ez rosszabb teljesítményt eredményezhet, mivel a cselekvések (valószínűleg) kevésbé jók, mint a jelenlegi politika. Kipróbálásuk nélkül azonban elképzelhető, hogy soha nem találja meg a lehetséges fejlesztéseket. Ezen túlmenően, ha a világ nem statikus, az ágensnek kutatást kell végeznie, hogy politikáját naprakészen tartsa. Tehát a tanuláshoz fel kell fedeznie, de ahhoz, hogy jól teljesítsen, ki kell aknáznia azt, amit már tud. E két dolog egyensúlyát „felfedezés-kiaknázás problémának” nevezik.

4. Statisztikai Minőségirányítás Megerősítéses Tanulással

A második fejeztben bemutattam a környezet működését, a harmadikban meg egy általános időbeli különbség tanuló algoritmust. Most az általunk kifejlesztett kód felépítését írom le, ahol ezt a kettőt implementáltuk és összekötöttük, különféle ellenőrzési és vizsgálati diagrammokkal kiegészítve.

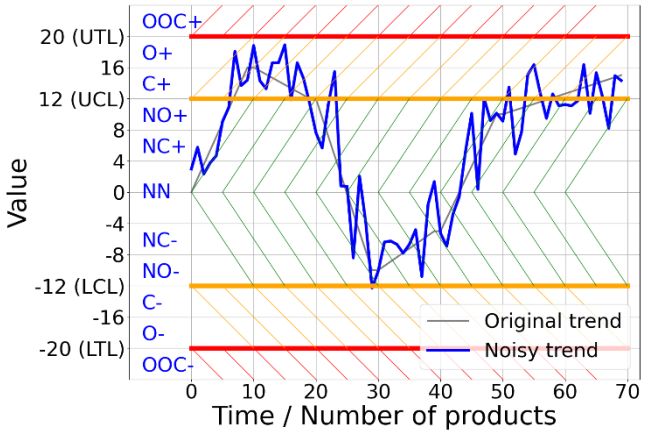
4.1. Első verzió

Legelőször a leírtak alapján megteremtettük a környezetet és felvettük a szakemberekkel előre megbeszélt akciókat, eseményeket és a hozzájuk tartozó büntetéseket, hatásokat. Ezután felépítettük az egyszerű ágenst és lehetővé tettük a környezettel való kommunikálást. Ezen felül, a sikeres termékek gyártása érdekében további jutalom kiegészítést adtunk a kódhoz.

A javasolt architektúrában a jutalmak a megtett intézkedések költségeiként vannak definiálva: negatívak, és minél nagyobb a cselekvési költség, annál alacsonyabb a jutalom. A szimulációban feltételezzük, hogy ha egy trendpont (termék) kilép a tűréshatárokon, az további büntetést von maga után (selejtes termék költsége). A büntetés nem egyenlő a két tűréshatáron kívül, ennek az az oka, hogy a felső hibahatáron kívüli termékek ”túl hosszúak”, amik valóban nem felhasználható termékek, viszont ki lehet őket még javítani, így a felső határon kívüliek költsége inkább javítási költség, míg az alsó hibahatáron kívüliek ”túl rövidek”, azokat már nem lehet kipótolni. Ha a gyártás az előírt tűréshatáron belül van, és nem történik intézkedés (tehát a „nincs akció” van kiválasztva), akkor az ágens jutalomként 0-t kap (a legjobb eset, ha nem jár büntetés).

4.2. Q tábla

A mély neurális hálók használata előtt a statisztikai minőségirányítást rejtett dinamikáját idősorként elemezték. Erre a célra eleinte egy egyszerű Q táblát használtak, fehér dobozos jellege miatt, ami azt jelenti, hogy a rejtett folyamatok láthatók. Mi a termékek adatpontjait folytonos értékekkel jellemezzük, ha ezt tárolnánk Q táblával, akkor a mérete exponenciálisan nőne. Megoldásként az szabályzó diagramm értéktartományát fix sávokra osztottuk, az ugyanabban a sávban lévő adatpontok ugyanazt a kvantált értéket/kódolást kapják. A felosztást tetszőleges végértékekkel (-20 és 20) tizenegy részre osztottuk, két szabályzáson kívüli sávra (-20 alatt, 20 felett), két szabályzó sávra ((-20,-12] között, [12, 20) között), és néhány ”normális” sávra (4.2.1 ábra).



4.2.1 ábra

Ez a megközelítés az ipari statisztikai minőségirányítási megközelítésekből lett átvéve. Ezek a számok teljesen önkényesek, más számok vagy határok is választhatók, hisz az értékek normalizálva vannak a valós tartományokra, így a megoldás kellően általános.

A Q táblában szereplő állapotokat tehát az elöbb leírtak alapján létre tudjuk hozni, ezt bemutatja a 4.2.2 ábra. Itt T a jelenlegi adatponthoz tartozó idő, BW az általunk kifejlesztett Backwards Window (Hátratekintő Ablak). Ezek szerint egy állapot nem csak a jelenlegi, hanem a BW-vel ezelöttig bezáródó adatponttól függ, ennek jelentősége az, hogy így egyfajta trendmegkülönböztető állapotvizsgálatot alkalmazunk (tegyük fel, hogy az utolsó BW adatpont sávja egyenként nőtt, akkor az nem ugyan az az állapot (értelemszerűen), mintha mind a BW egyazon sávban lenne). A mínusz és plusz jelek a sávok szimmetriája miatt lévő megkülönböztetés végett van ott.

States			
$T - BW_S$	...	$T - 1_S$	T_S
C-	...	NC-	NN
NN	...	NO+	NC+

4.2.2 ábra

A Q táblában az egyes állapotokhoz tartozó lehetséges akciók értékeit a következőképpen tároljuk: A különálló akciótáblázatban, ahol az oszlopok a lehetséges akciók, az alattuk lévő

$V_{(i,j)}$ értékek pedig az akciók (j) becsült értke , a hozzá tartozó állapot (i, sor) tekintetében, ezek az ún. Q értékek (4.2.3 ábra) [1].

States				Production Actions			
$T - BW_S$	...	$T - 1_S$	T_S	A_1	A_2	...	A_N
C-	...	NC-	NN	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$
NN	...	NO+	NC+	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$

4.2.3 ábra

Ezt a következőképpen kell leolvasni: egy állapotot választunk a sorokból, aztán abból a sorból egy akciót az oszlopokból. Az állapotok kvantált értékekből állnak, hosszuk attól függ, hogy hány korábbi értéket veszünk figyelembe az akcióválasztásnál (BW).

Amikor a szakértők, kezelők vagy egy vezérlőrendszer a legjobb termelési beavatkozást, azaz a legjobb akciót keresi, akkor ennek a sornak a legnagyobb értéke a válasz. Az alkalmazott, jól ismert Q táblás ábrázolás szerint a termeléssel kapcsolatos ismereteket a V(i,j) értékekben tároljuk, ezt később egy (vagy több) regressziós technikával, például mély neurális hálózatokkal helyettesíthetjük.

Ezeket kiegészítve gyártási környezetben döntő fontosságú annak figyelembe vétele, hogy az adott gyártási folyamat irányítása érdekében mik voltak az utolsó akciók, mert nem érdemes rövid időn belül ugyanazt (pl. költséges) az akciót megismételni. Ennek érdekében egy állapothoz eltároljuk az utolsó ”n” lépésnél végrehajtott akciókat is (4.2.4 ábra).

States							Production Actions			
$T - BW_S$	...	$T - 1_S$	T_S	$T - 3_A$	$T - 2_A$	$T - 1_A$	A_1	A_2	...	A_N
C-	...	NC-	NN	A_3	A_5	A_5	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$
NN	...	NO+	NC+	A_6	A_N	A_8	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$
				...						

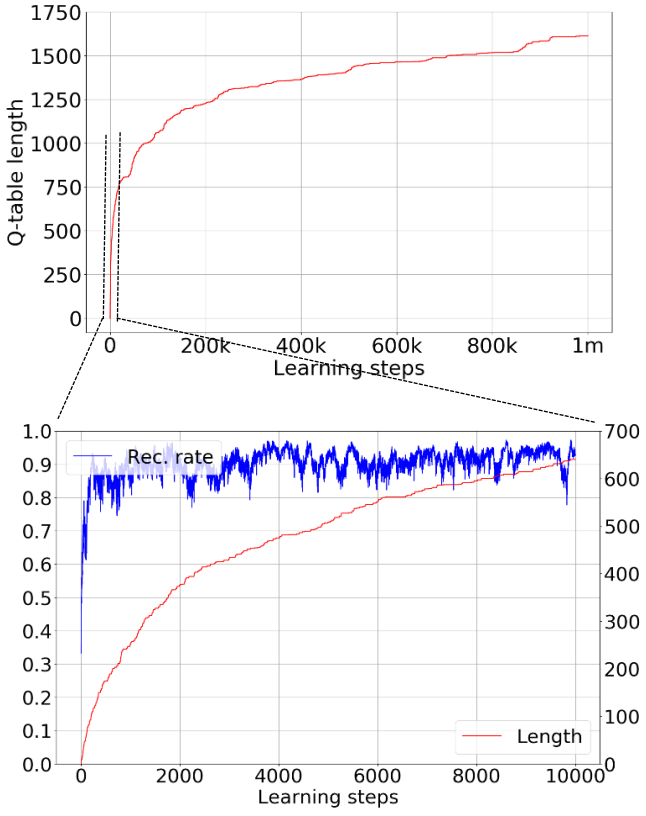
4.2.4 ábra

Egy másik, hasonló módszer is kipróbáltunk és ez van a jelenlegi implementációban. A lényege az, hogy az utolsó ”n” akció helyett minden akcióhoz tartozik egy quantizált érték tartozik, hogy hány lépéssel ezelőtt volt utoljára választva (4.2.5 ábra).

States							Production Actions			
$T-BW_S$	...	$T-1_S$	T_S	A_1	...	A_N	A_1	A_2	...	A_N
C-	...	NC-	NN	0-1	...	4-90	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$
NN	...	NO+	NC+	1-10	...	0-2	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$
				...						

4.2.5 ábra

A Q táblás módszerekhez az előre létrehozott Q tábla értékei lépésről lépésre változnak. Q tábla fő problémája a mérete. Például egy A oszlopokkal rendelkező tábla esetén, ahol minden oszlop B különböző értéket vesz fel, a táblázat mérete elérheti a B^A sort. Nagy A és B esetén a tábla generálása, tárolása és kezelése problémákat okozhat, ráadásul az üres táblához használat előtt nem szükséges memóriát lefoglalni. Ezért bevezettünk egy technikát, az úgynevezett dinamikus Q táblát, ahol csak annyi memóriát foglalnak le, amennyi szükséges, és csak akkor, amikor szükséges. A javasolt koncepció alapján a tanulás elején a Q tábla üres. Amikor a tanuló algoritmus új állapotot ér el, hozzáadja azt egy listához, amely a Q tábla sorait reprezentálja, és az akciók kezdeti értékei véletlenszerűen kerülnek kiválasztásra. Ha pedig az aktuális állapot már szerepel a Q táblában, akkor a kiválasztott akciók értéke a frissítési szabály szerint frissül. Kezdetben, amikor az ágens leginkább a környezetét fedezi fel, gyakran találkozik olyan állapotokkal, amelyekben korábban nem volt, ezért azok felkerülnek a Q tábla végére. Sok felfedezés után viszont egyre sűrűbben találkozik olyan állapottal, amiben viszont már volt, így a Q tábla növekedése lényegesen lelassul, és csak a benne lévő értékeket változtatja. A táblázat hossza logaritmikusan növekszik, ahogy az a 4.2.6-os ábra felső részén látható.



4.2.6 ábra

Az alsó képen lévő kék vonal a meglátogatott állapot felismerési arányát mutatja, azaz, hogy megtalálható-e a Q táblában. Ebből az látszik, hogy 2000 tanuló lépés után a Q tábla mérete nő, míg a felismerési arány ugyan azon a (kellően magas) értéken stagnál, ennél tovább tanulni már nem ”hasznos”. Ez egy nagyon fontos eredmény, amely bizonyítja, hogy a Q táblának van racionális korlátja, ezen túl jelentősen nő a méret, a számítási idő (állapotok megkeresésére) és az egyéb teljesítményigény (pl. memória), de nem hoz értékes plusz tudást az adott statisztikai minőségirányítási feladathoz. Következésképpen a javasolt dinamikus Q tábla megoldással az informatikai háttérkövetelmények korlátozhatók és kontroll alatt tarthatók, ami eliminálja a mesterséges intelligencia egyik fő problémáját.

4.3. Kipróbált felfedezési módok

A tanulás során a legnépszerűbb ε-mohó algoritmust alkalmazzuk, amelyben ε szabályozza a felfedezés és a kiaknázás arányát. 0 érték azt jelenti, hogy csak kiaknázás van, az 1 pedig teljes felfedezés [1]. Különféle stratégiák léteznek az ε értékének beállítására a tanulás során, jellemzően nagy értékkel kezdődik a széleskörű felfedezés érdekében, és idővel csökken a kiaknázás érdekében; ahogyan az ágens felfedez és egyre pontosabb képe van a környezet működéséről, inkább kiaknázza a megszerzett tudást. Ez egy igazán értékes tulajdonsága az megerősítéses tanulásnak, de másrészt egy további paraméter, amelyet definiálni és ellenőrizni kell. Ezt az értéket előre beállítottan lecsengő értéknek is be lehetne állítani, de mi önszabályzó megoldást dolgoztunk ki az ε dinamikus beállítására, ráadásul maga az felfedező ágens által. Az új megoldás lényege az, hogy kiterjesztjük a Q táblát a megfelelő ε érték kiválasztására szolgáló táblázattal (4.3.1 ábra). Minden (előre meghatározott

States							Production Actions				Epsilon Actions			
$T-BW_S$	...	$T-1_S$	T_S	A_1	...	A_N	A_1	A_2	...	A_N	E_1	E_2	...	E_M
C-	...	NC-	NN	0-1	...	4-90	$V_{1,1}$	$V_{1,2}$	...	$V_{1,N}$	$E_{1,1}$	$E_{1,2}$	...	$E_{1,M}$
NN	...	NO+	NC+	1-10	...	0-2	$V_{2,1}$	$V_{2,2}$	...	$V_{2,N}$	$E_{2,1}$	$E_{2,2}$	...	$E_{2,M}$
				...										

4.3.1 ábra

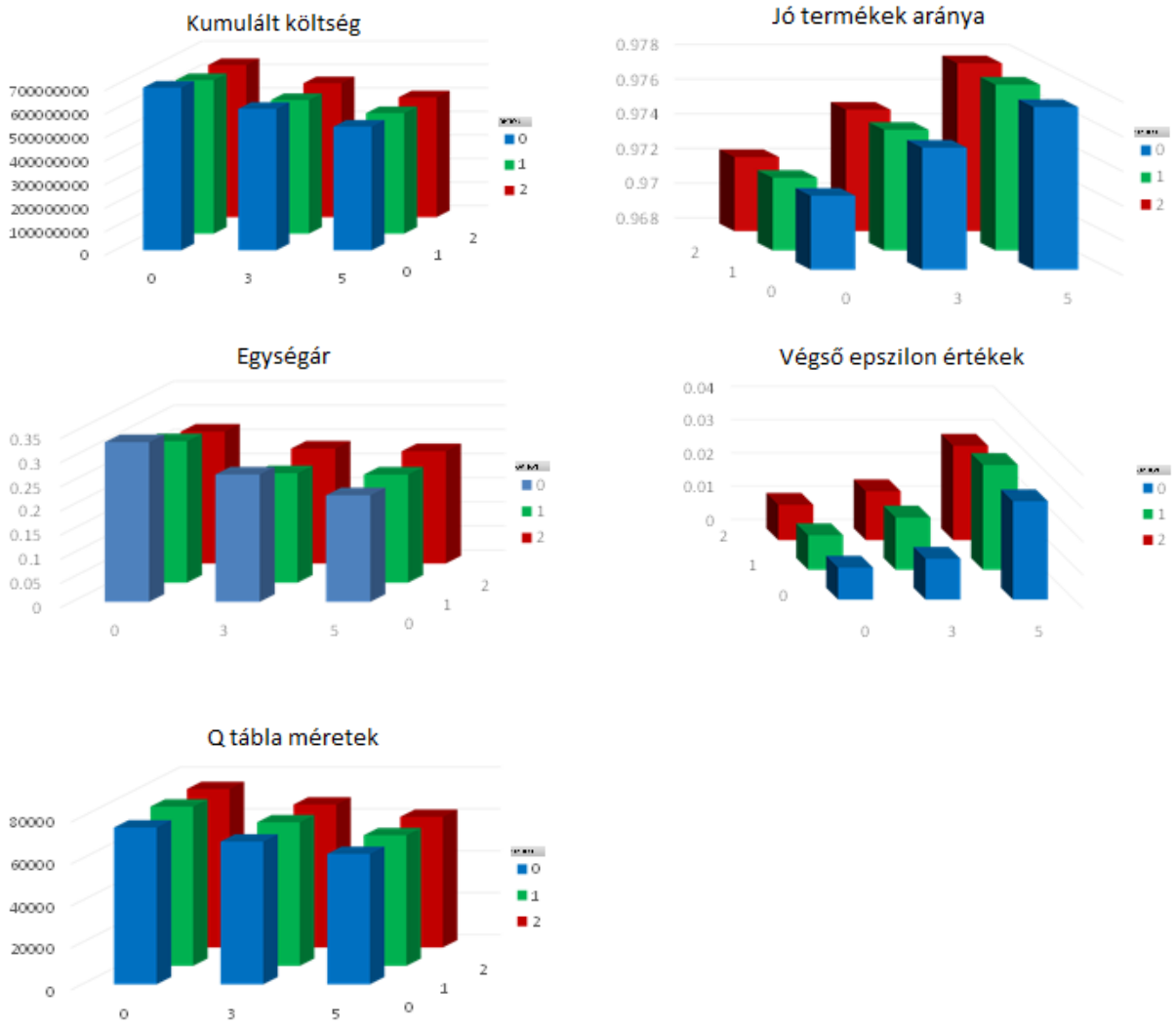
számosságú és értékű) ε-hoz (E_j) tartozik egy érték, $E_{(i,j)}$. Miután megvan az ε, ez alapján választunk akciót, mint az eddigiekben. A ε értékeihez tartozó frissítési szabály nagyon hasonlít az akciók értékeihez tartozó szabályra. Ennek a stratégiának a hasznosságát különböző kísérletek is igazolták, az alkalmazott megerősítéses tanulás algoritmus az epizódok során folyamatosan szabályozta saját felfedezési arányát. Különösen ígéretes az a viselkedés, amelyet egy váratlan, külső esemény bekövetkeztekor tapasztaltunk, ekkor az ágens automatikusan magasabb feltárási arányt állít be az újszerű állapot kezelésére, míg más esetekben kicsiben, tehát a szinte teljes kiaknázás tartományában tartja.

Meghatároztuk az ágens felfedezés-kiaknázás politikáját, azonban amikor felfedezünk, se kell az összes lehetséges akciót egyenlő eséllyel kipróbálni. A következő rész ennek az ötletnek a részletes vizsgálatával foglalkozik.

Három különböző típusú felfedezési lehetőséget határoztunk meg, elemeztünk és értékeltünk: Az első opciónál, ha az ágens úgy dönt, hogy felfedezi a környezetét, az akció véletlenszerűen kerül kiválasztásra az összes lehetséges akció közül, egyenletes eloszlással. A második opció viszont már figyelembe veszi az akciók jelenlegi értékét a Q táblában, és ezeknek az arányában dönt. A harmadik opció az egyes akciók által kapott, ismert jutalmak arányában dönt. Ez a harmadik pusztán tesztelési szempontból érdekes, hiszen az ágens nem tudja előre a megkapott jutalom értékét, ezen kívül az adat egy része a Q táblában van tárolva, így ez inkább hasonlít a második opcióhoz. Nincs módszer a legjobb felfedezőfüggvény megtalálására, ezért tesztelnünk kell őket és az összehasonlítás alapján dönteni, amit a későbbiekben bemutatok.

A termelési akciónál az aktuális ε szabályozza a feltárás-kiaknázás arányt, azonban a jövőbeli ε érték kiválasztásához csak a kiaknázást határoztuk meg (ez a megoldás teljesített a legjobban). Amikor viszont az ágens először találkozik egy állapottal, akkor választania kell egy ε kezdőértéket. Ez az ε határozza meg az állapot pillanatnyi felfedező-kiaknázás arányát. A választható ε értékek: 0; 0.05; 0.15; 0.25; 0.5; 1. Így ebben az új állapotban a felfedés mértékéhez is tartozik egyfajta felfedezhetőség. Nem lehet tudni a legjobb kezdőértéket, de a későbbiekben ezt is tesztelni fogjuk. Észrevehető, hogy az ε lehetséges értékei nem oszlanak meg egyenlően, mert a tapasztalatok azt tükrözték, hogy a kiaknázás aránya sokkal nagyobb, mint a felfedezésé, következésképpen kisebb ε értékekre nagyobb szükség van, mint nagyobbakra.

Átfogó kísérletet végeztünk a legjobb felfedezésszabályozás és a kapcsolódó optimális kezdeti felfedezési szint kiválasztására. Az alábbi kísérleteket futtatuk le a felfedezés kezdőértékével kapcsolatban (0-s számú tesztelés: 0.0; 3-as számú tesztelés: 0.25; 5-ös számú tesztelés: 1.0): Minden kombinációnál mértük a végső kumulált költségeket, a jó termékek arányát, a termék egységárát, az végső önszabályzott felfedezés-kiaknázás arányt és a Q tábla méretét. A kísérleti eredményeket a 4.3.2-es ábra mutatja.

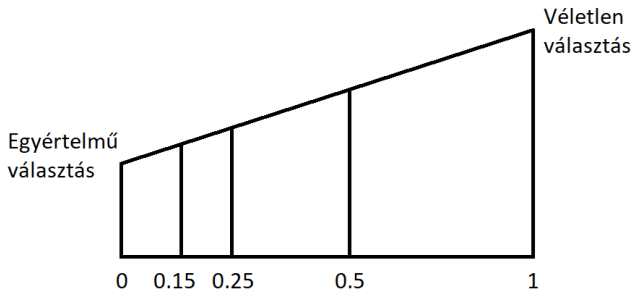


4.3.2 ábra

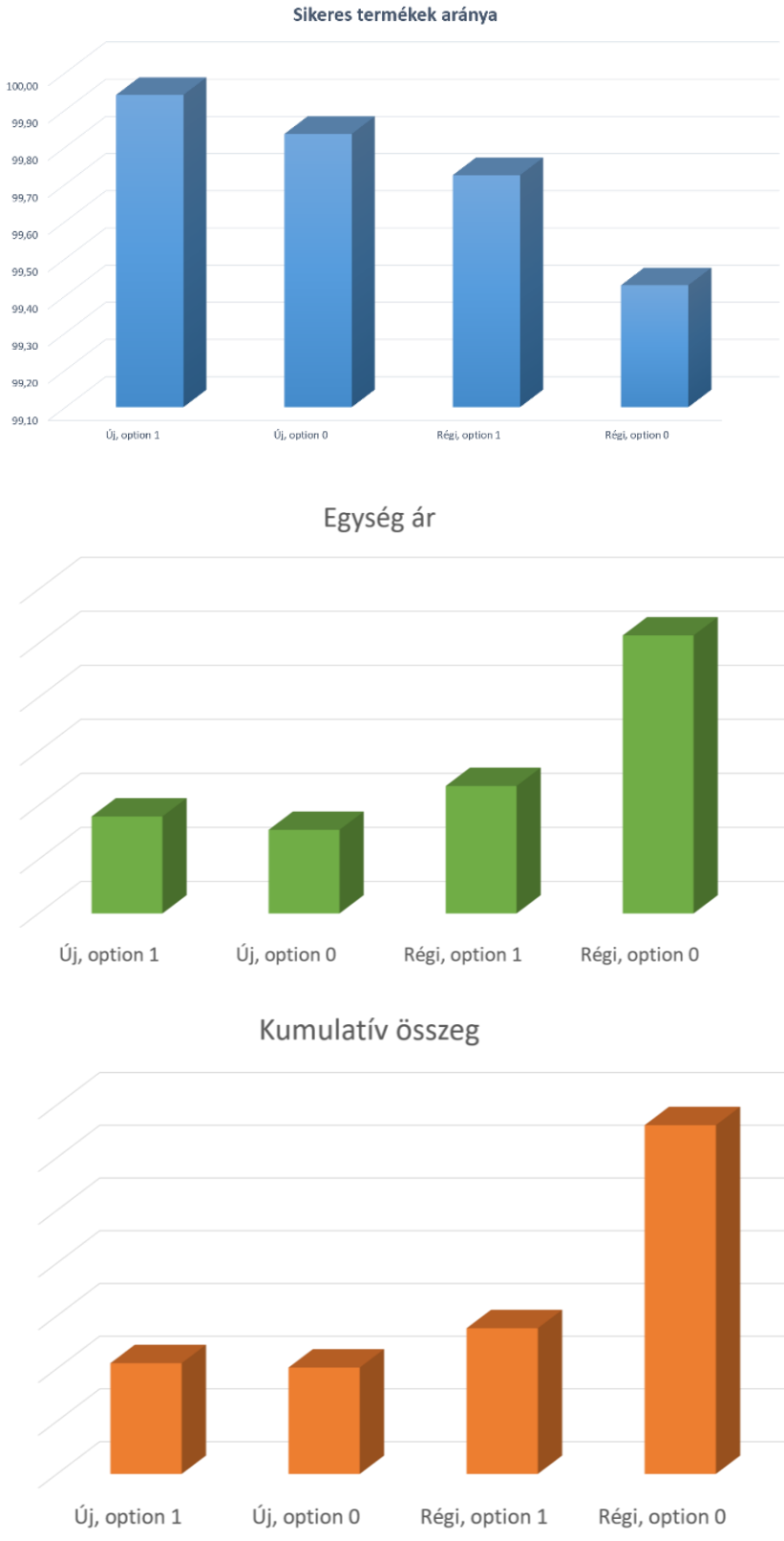
Az adatokból az látható, hogy a vizsgáltak közül a második opcióhoz tartozó felfedezési szabályzási szabály az optimális (az akció az aktuális állapothoz tartozó Q tábla akció értékeivel arányos véletlenszerű választás során választódik ki) és a kezdeti feltárási szint az 5-ös számú tesztelés, amihez az ε = 1.0 érték tartozik, mint teljes felfedezés!

A tesztek futtatása óta, egy ezeknél mégjobbnak bizonyuló módszert fedeztünk fel, aminek a lényege az epszilon kezdőértékeinek a beállításában rejlik. Minden egyes epszlinhoz beállítjuk a hozzá tartozó várható értéket, a felfedezés-kiaknázás aránnyal és a felfedezésnél lévő akciók kiválasztási esélyének és várható jutalmának függvényében (4.3.3 ábra). Tehát kétféle felfedezést (régi: véletlen epszilon kezdőértékek és új: várható jutalom szerinti epszilon kezdő értékek) és kétféle felfedezésen belüli akcióválasztási módszert (option 0: egyenlő esélyű akció és option 1: jutalommal fordítottan arányos esélyű akció) hasonlítottunknk össze (4.3.4 ábra).

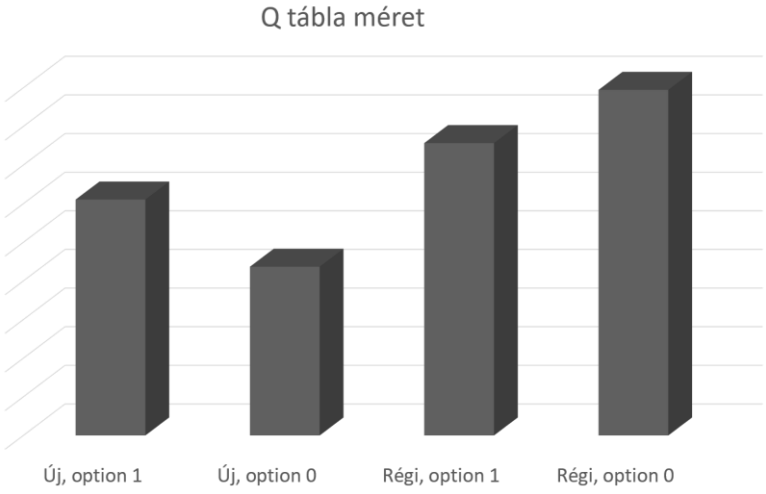
Ezeken az adtokon az látszik, hogy a várható jutalom szerinti epszilon kezdő értékekkel és a felfedezésnél a jutalommal fordítottan való akcióválasztással szinte ugyan azon a költségszinten egy jóval magasabb sikeres termék arányt érünk el. Itt jól látható, hogy csak egy százalék tört részével több a sikeres termék, amit viszont úgy is megfogalmazhatunk, hogy nagyjából feleannyi selejtes termék lesz az új módszerrel, mint ezelőtt.



4.3.3 ábra


Sikeres termékek aránya
100,00
99,90
99,80
99,70
99,60
99,50
99,40
99,30
99,20
99,10
Új, option 1
Új, option 0
Régi, option 1
Régi, option 0
Egység ár
Új, option 1
Új, option 0
Régi, option 1
Régi, option 0
Kumulatív összeg
Új, option 1
Új, option 0
Régi, option 1
Régi, option 0

4.3.4 ábra

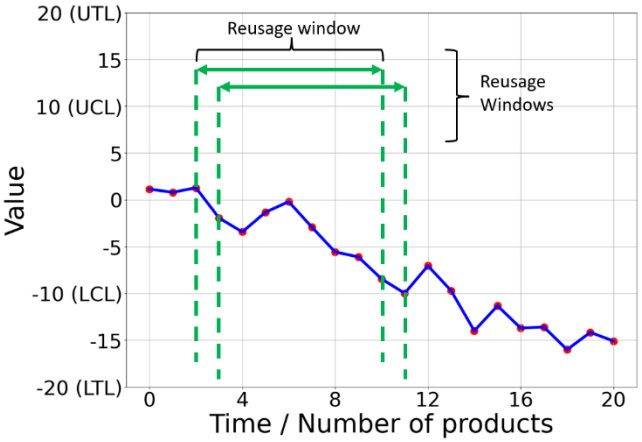


4.3.4 ábra

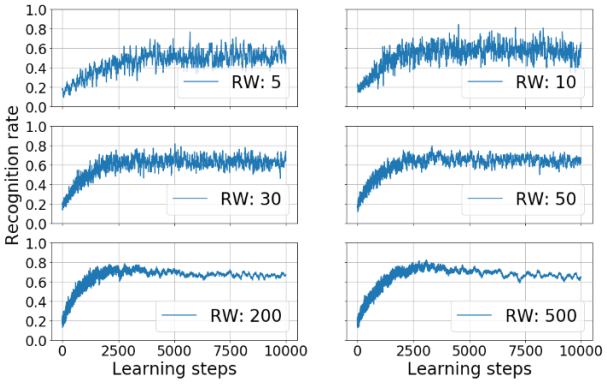
4.4. Bevezetett módszerek, MW, RW

A gyártási környezetben minden egyes mért érték, számos forrásból adódó, erősen egymáshoz kapcsolódo költséget tartalmaz, pl. a felszerelés költsége, a személyzet képzése, a mérési tevékenységek elvégzése, az adatgyűjtés informatikai hátterének kialakítása, az informatikai kommunikáció és tárolást, a méréseket lebonyolító szoftver, a mérőeszközök folyamatos újrakalibrálása, karbantartása stb. Következésképpen a gyártásban mért értékek magas üzleti és műszaki értéket foglalnak magukban, ezért ezeket a lehetőségekhez mérten ki kell használni. A mai termelésirányítási környezetekben ez az ideális helyzet még messze nem közelíthető meg, az adatvagyon jelentősen meghaladja a felhasználását és kiaknázását. Az előző fejezetekben bemutatott újszerű módszerek fő problémája, hogy hiányzik az adatok újrafelhasználtsága, hasonlóan a kapcsolódó szakirodalomban szereplő esetekhez. A gyártásban az ilyen idősoros mérésekben, minden mérés egy folyamat során előállított termék adata, amely költséges, vagy költségesnek tartott legalább az extrém költségnyomás miatt. Következésképpen az összes mérési érték egyszeri felhasználása nagyon pazarlónak tűnik. Ebből kifolyólag kívánatos a mérési értékek többször felhasználni, ezért vezettük be a Reusage Window (Újrahasználó Ablak, RW) koncepciót, amely meghatározza, hogy egy egyedi mért érték hányszor lesz újra felhasználva az általunk használt megerősítéses tanuló algoritmus szerint. A következőkben ezt részletesen kibontom.

Az eredeti koncepcióval ellentétben, ahol az ágens csak egyszer járja végig az idősort, most először kiválasztunk egy intervallumot, amelynek hossza RW, és az ágens végimegy ezen, és mintavételezi, kvantifikálja az állapotokat. Ez egy tanulási iteráció. Ezután az RW egy lépéssel előrébb lép az idősoron, és újraindul, egészen addig, amíg az RW el nem eléri az idősor tényleges végét (4.4.1 ábra). Ez azt jelenti, hogy egy adatot annyiszor használunk fel, ameddig az RW hossza. Ezeken az újrahasznált intervallumon belül az ágens átmegy, és minden helyzetet feldolgoz a feljebb említett epszilon mohó algoritmusban leírtak szerint. Így végül minden mért adatot RW-szer dolgoztunk fel [51]. Az előzetes teszteredmények azt mutatják, hogy az MW-nek van optimális értéke, amint azt a 4.4.2-es ábra is kiemeli.

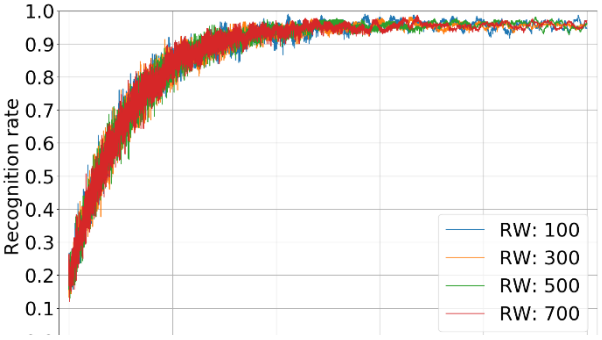


4.4.1 ábra



4.4.2 ábra

Az ágensek reakcióinak teljesítményét minden tanulási iteráció után a kapott jutalmak, a jó termékek aránya, az egységköltség és az önszabályzott epszilonok átlagai alapján értékeljük. A Measurment Window (Mérési Ablak, MW) fogalmát már korábban bevezették [51], hogy lehetővé tegye a különböző RW-vel rendelkező folyamatszabályozási algoritmusok igazságos összehasonlítást megvalósítandóan. A különböző ugynökök teljesítményparaméterei nagyon ingadozók és megbízhatatlanok voltak, így definiáltak egy független időablakot, amely a legutóbbi múltbeli adatok számát adja meg az ágens tényleges teljesítményének értékelésének alapjaként. A különböző RW de azonos MW



4.4.3 ábra

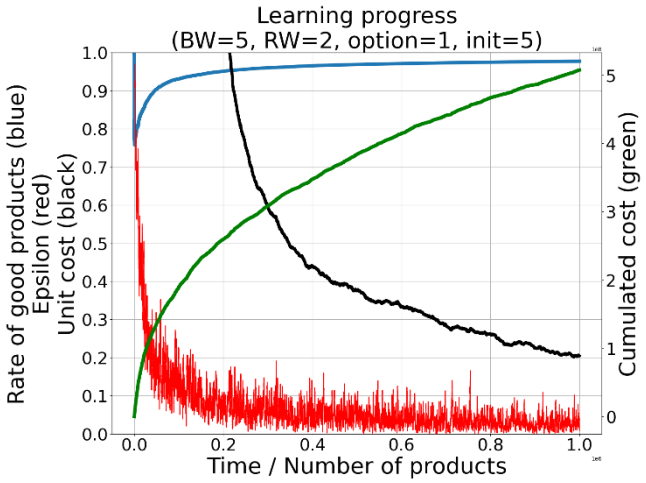
paraméteres futtatások eredményeit a 4.4.3-mas ábra mutatja meg. Ezzel a módszerrel a teljesítmények egyenlőek, míg az MW használata nélkül különbözőek lettek volna. A végleges előzeti kutatási eredmények azt tükrözték, hogy az RW és az MW is optimális értékkel rendelkezik, így a gyakorlati kísérletekben meta-paraméterként adhatók meg.

5. Eredmények

A rendszer teljesítménymutatóit a kumulált előállítási költségen, a jó termékek arányán (meghibásodás nélkül, a tűréshatáron belül), a termék egységköltségén (a kumulált költség származéka) és az (ágens által) önszabályzó felfedezési arányon keresztül mutatjuk be. Az 5.1-es ábra a folyamatirányítás hatékonyságának folyamatos növekedését tükrözi, köszönhetően az ágens megerősítéses tanulási képességeinek. Az epizódok során folyamatosan növekszik az ágens ismerete, így egyre jobb lett a termelésirányítási akciók kiválasztása, az ágens alkalmazkodik a különféle, változó termelési körülményekhez. A folyamatosan tanuló és teljesítő ágens gyártáshoz kapcsolódó teljesítmény-összehasonlítása a kiválasztott gyártási akció gyakoriságok megoszlásának elemzésével, valamint az egységdíj és a jó termékek arányának elemzésével valósul meg, a két utóbbi az ágens két legfontosabb teljesítménymutatói.

A tanuló ágens termelési teljesítménye akkor tekinthető kielégítőnek, ha az egységdíj átlaga (fekete vonal) és a jó termékek aránya (kék vonal) állandó (egyenes), tehát ha a mozgóátlaguk egy előírt (kis) tűréstartományon belül vannak, és az önszabályzó epszilon pedig egy előírt (kis) tűréshatárnál kisebb.

Ahogy az 5.1-es ábrán látható, az ε arány (piros vonal) szintje folyamatosan csökken, tükrözve, hogy a tanulás során az ágensnek folyamatosan csökken a saját felfedezési igénye. Bizonyos számú tanulási lépés után valóban kicsi, de nullánál nagyobb szinten tartja, ami azt jelenti, hogy az ágens saját tudását használja fel a lehető legjobb termelési akció kiválasztására szinte minden esetben, és így nagyon ritka, hogy felfedezi a környezetet. A minimális, de nem nulla érték tartása optimális a nem állandó környezetkben.



5.1 ábra

Következésképpen, az ágens gyártási folyamata közel van az optimumhoz, amit az egységköltség és a jó termékek szintjének állandó görbéi is mutatnak. Miután ezt a teljesítményszintet elértük, a szimulált gyártási akciók gyakoriságát összehasonlítottuk a valós gyártási környezetben előforduló gyakoriságukkal, hogy a javasolt koncepció teljesítményét számszerűen megmérjük és validáljuk.

A legfontosabb szempont a kiválasztott „nincs akció" típusú akció aránya volt, mivel a valóságban a legyártott termékek többségénél a gyártás automatikusan, külső beavatkozás nélkül zajlik, így a valós „nincs akció" akció aránya nagyon közel van az 1-hez. Ugyanezt a szintet érte el a mi algoritmusunk is, ráadásul a legjobb esetekben ez az arány még magasabb is volt, mint a valós, mintegy 10-30%-kal kevesebb gyártási beavatkozást hajtott végre, mint amit a valós gyártóműhelyben szoktak.

Egy nagyon fontos különbség a valós gyártásmenethez képes az, hogy ott csak akkor avatkoznak be, ha hibát látnak vagy ha egy mérési eredmény azt megindikolja. A mérések a szimulációban minden egyet terméken megtörténnek, a tényleges gyártósoron ez csak fix időnként, több termék után történik meg, ez a már leírt gazdasági okoknak köszönhető. Ebből kifolyólag az algoritmust használva a felügyeleti szint is szignifikánsan magasabb.

Végül bebizonyosodott, hogy a bemutatott ágens képes egyszerre tanulni és teljesíteni, viselkedését a környezeti változásokhoz, zavarokhoz igazítani, valamint a minimális egységköltséget és a jó termékek maximális arányát elérni. Következésképpen a gyártásban a megerősítéses tanuláson alapuló statisztikai minőségirányítás megvalósítására bevezetett koncepció helyesen szerepelt.

Az alábbi teszteredmények sokszor napokig futottak, a tárhelyigény és a számítási költségek túl nagyon voltak egy egymilliós nagyságú szimulációhoz. Az utóbbi időben ezen a téren is történt fejlesztés. Pontosabb összehasonlítás érdekében egy 250.000 hosszúságú tesztet vizsgáltunk, maximálisan egy órát hagyva rá. A régebbi verzió egy teljes óra alatt csak 134.000 lépést hajtott végre, míg az új mind a 250.000-et leszimulálta 87 másodperc alatt (0.02 óra). A sebességnövekedés fő öka a tárhelygazdálkodásra vezethető vissza. Mivel minden lépést eltárolunk nagyon fontos, hogy csakis a sokféle vizsgálathoz közvetetten szükséges adatokat mentsük el.

Még fontos továbbfejlesztés, hogy a már lefutott ágens politikáját el lehet menteni, az összes környezeti előzménnyel együtt, és be is lehet tölteni egy új futtatásba, ahol akár

különböző értékekkel tudjuk folytatni a tanulást. Mivel a megerősítéses tanulás hamar hozzászokik az új környezethez, hasonló eredményekre számítunk betöltés után is. Több kísértlet bizonyítja, hogy az ágens tényleg betanul és sokkal rosszab körülmények között is képes majdnem ugyan olyan jó eredményt elérni.

6. Jövő

Az együttműködő Opel Szentgotthárd Kft.-vel folytatott kutatást a következő szintre akarjuk emelni, ami pedig a valós életbeli alkalmazás lenne. Ehhez természetesen sok fejlesztés kell. Kompatibilissé tenni a saját kódot, ezután először még csak hozzákötni a gyártósori rendszerhez. Az ágens először csak javaslatokat tenne a szakembereknek, aztán ha minden tökéletesen megy a végén teljesen önállóan szabályozza a gyártási folyamatokat.

Köszönetnyílvánítás

Először is szeretném megköszönni Dr. Viharos Zsolt Jánosnak, hogy felkérhettem TDK Konzulensemnek, és a dolgozat írása alatt nyújtott segítséget.

Jakab Richárdnak és Dr. Viharos Zsolt Jánosnak, akik a kutatást elkezdték, és megengedték, hogy az abból írt cikket felhasználjam a dolgozatomhoz [51].

A publikációban szereplő kutatást, amelyet Számítástechnikai és Automatizálási Kutatóintézet valósított meg, az Innovációs és Technológiai Minisztérium és a Nemzeti Kutatási, Fejlesztési és Innovációs Hivatal támogatta a Mesterséges Intelligencia Nemzeti Laboratórium keretében.

Opel Szentgotthárd Kft. amely folyamatos, rendszeres együttműködést tart velünk.

The research in this paper was (partially) supported by the European Commission through the H2020 project EPIC (https://www.centre-epic.eu/) under grant No. 739592.

Irodalomjegyzék

[1] **Barto, A.G., Sutton, R.S., & Brouwer, P.**: Associative search network: A reinforcement learning associative memory, *Biological Cybernetics*, Vol. 40. 1981, pp. 201-211.

[2] **Bouazza, W.; Sallez, Y.; Beldjilali, B.**: A distributed approach solving partially flexible job-shop scheduling problem with a Q-learning effect, *IFAC PapersOnline 50-1*, 2017, p. 15890–15895.

[3] **Khader, N.; Yoon, S. W.:** Online control of stencil printing parameters using reinforcement learning approach, *Procedia Manufacturing 17*, 2018, pp. 94–101

[4] **Wang, Y-C.; Uscher, J. M.:** Application of reinforcement learning for agent-based production scheduling, *Engineering Applications of Artificial Intelligence* 18, 2005, pp. 73–82.

[5] **Waschneck, B.; Reichstaller, A.; Belzner, L.; Altenmüller, T.; Bauernhansl, T.; Knapp, A.; Kyek, A.;** Optimization of global production scheduling with deep reinforcement learning, *Procedia CIRP*, 72, 2018, pp. 1264–1269.

[6] **Schneckenreither M.; Haeussler S.:** Reinforcement Learning Methods for Operations Research Applications: The Order Release Problem. *In: Nicosia G., Pardalos P., Giu rida G., Umeton R., Sciacca V. (eds) Machine Learning, Optimization, and Data Science, Part of the Lecture Notes in Computer Science book series (LNCS, volume 11331)*, 2019, pp. 545-559.

[7] **Kuhnle, Al.; Schäfer, L.; Stricker, N.; Lanza, G.:** Design, Implementation and Evaluation of Reinforcement Learning for an Adaptive Order Dispatching in Job Shop Manufacturing Systems, *Procedia CIRP*, 81, 2019, 234–239.

[8] **Kuhnle, A.; Röohrig, N.; Lanza, G.:** Autonomous order dispatching in the semiconductor industry using reinforcement learning. *Procedia CIRP*, 79, 2018. pp. 391–396.

[9] **Kardos, Cs.; Laflamme, C.; Gallina, V.; Sihn, W.**: Dynamic scheduling in a job-shop production system with reinforcement learning, *Procedia CIRP, 8th CIRP Conference of Assembly Technology and Systems*, 29 Sept. – 1. Oct., Athens, Greece, 2020., in print.

[10] **Qu S., Wang, J., Govil, S., Leckie, J. O.:** Optimized Adaptive Scheduling of a Manufacturing Process System with Multi-SkillWorkforce and Multiple Machine Types: An Ontology-Based, Multi-Agent Reinforcement Learning Approach, *Procedia CIRP*, 57, 2016, pp. 55–60.

[11] **Nair, A.; McGrew, B.; Andrychowicz, M.; Zaremba, W.; Abbeel, P.**: Overcoming Exploration in Reinforcement Learning with Demonstrations, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 6292-6299.

[12] **Plappert, M.; Andrychowicz, M.; Ray, A.; McGrew, B.; Baker, B.; Powell, G.; Schneider, J.; Tobin, J.; Chociej, M.; Welinder, P.; Kumar, V.; Zaremba, W.:** Multi-Goal Reinforcement Learning: Challenging Robotics Environments and Request for Research, ArXiv, (2018), abs/1802.09464.

[13] **Zhu, Y.; Wang, Z.; Merel, J.; Rusu, A.; Erez, T.; Cabi, S.; Tunyasuvunakool, S.; Kramár, J.; Hadsell, R.; Freitas, N.; Heess, N.:** Reinforcement and Imitation Learning for Diverse Visuomotor Skills, *Proceedings of Robotics: Science and Systems*, Pittsburgh, Pennsylvania, 2018, 10 p.

[14] **Kahn, G.; Villaflor, A.; Ding, B.; Abbeel, P.; Levine, S.:** Self-Supervised Deep Reinforcement Learning with Generalized Computation Graphs for Robot Navigation, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 5129-5136.

[15] **Long, P.; Fan, T.; Liao, X.; Liu, W.; Zhang, H.; Pan, J.:** Towards Optimally Decentralized Multi-Robot Collision Avoidance via Deep Reinforcement Learning, *2018 IEEE International Conference on Robotics and Automation (ICRA)*, Brisbane, QLD, 2018, pp. 6252-6259.

[16] **Johannink, T.; Bahl, S.; Nair, A.; Luo, J.; Kumar, A.; Loskyll, M.; Ojea, J.A.; Solowjow, E.; Levine, S.:** Residual Reinforcement Learning for Robot Control, *2019 International Conference on Robotics and Automation (ICRA)*, Montreal, QC, Canada, 2019, pp. 6023-6029.

[17] **Popescu, A.**: *Electrical machines and drives*, Politehnium, Iasi, 2011, p. 234.

[18] **Curtley, M, Prince, O.A.**: A synchronous detector with improved parameters, *Proceeding of the International Symposium on Circuits and Systems*, Barcelona, July 15-17, 2009, pp. 255-260.

[19] **Ranaee, V.; Ebrahimzadeh, A.**: Control chart pattern recognition using a novel hybrid intelligent method, *Applied Soft Computing*, Vol.11, 2011., pp. 2676–2686.

[20] **Lavangnananda, K.; Khamchai, S.**: Capability of Control Chart Patterns Classifiers on Various Noise Levels, *Procedia Computer Science*, Vol. 69, 2015, pp. 26–35.

[21] **Viharos, Zs. J.; Csanaki, J.; Nacsa, J.; Edelényi, M.; Péntek, Cs.; Kis, K. B.; Fodor, Á.; Csempesz, J.**: Production trend identification and forecast for shop-floor business intelligence, *ACTA IMEKO, The Open Access e-Journal of the International Measurement Confederation (IMEKO)*, Vol. 5. No. 4., 2016., pp. 49-55. ISSN: 2221-870X

[22] **Köksal, G.; Batmaz, I.; Testik, M. C.**: A review of data mining applications for quality improvement in manufacturing industry, *Expert Systems with Applications*, Elsevier, Vol. 38., 2011, pp. 13448–13467.

[23] **El-Midany, T. T.; El-Baz, M. A.; Abd-Elwahed, M.S.**: A proposed framework for control chart pattern recognition in multivariate process using artificial neural networks, *Expert Systems with Applications*, Vol. 37, 2010., pp.1035–1042.

[24] **Pelegrina, G. D.; Duarte, L. T.; Jutten, C.**: Blind source separation and feature extraction in concurrent control charts pattern recognition: Novel analyses and a comparison of different methods, *Computers & Industrial Engineering*, Vol. 92., 2015., pp. 105-114.

[25] **Gutierrez, H. De la T.; Pham, D.T.**: Estimation and generation of training patterns for control chart pattern recognition, *Computers & Industrial Engineering*, Vol. 95, 2016., pp. 72-82.

[26] **Yang, W.-A.; Zhou, W.; Liao, W.; Guo, Y.**: Identification and quantification of concurrent control chart patterns using extreme-point symmetric mode decomposition and extremelearning machines, *Neurocomputing*, Vol. 147, 2015, pp. 260-270.

[27] **Motorcu, A. R.; Güllü, A.**: Statistical process control in machining, a case study for machine tool capability and process capability, *Materials and Design*, Vol. 27, 2006., pp.364–372.

[28] **Huybrechts, T.; Mertens, K.; De Baerdemaeker, J.; De Ketelaere, B.; Saeys, W.**: Early warnings from automatic milk yield monitoring with online synergistic control, *American Dairy Science Association*, Vol.v97, 2014, pp. 3371-3381.

[29] **Viharos, Zs. J.; Monostori, L.**: Optimization of process chains by artificial neural networks and genetic algorithms using quality control charts, *Proceedings of Danube - Adria Association for Automation and Metrology*, Dubrovnik, 1997., pp. 353-354.

[30] **Viharos, Zs. J.; Kis K. B.**: Survey on Neuro-Fuzzy Systems and their Applications in Technical Diagnostics and Measurement, *Measurement*, Vol. 67., 2015., pp. 126-136.

[31] **Móricz, L.; Viharos, Zs. J.; Németh, A.; Szépligeti, A.; Büki, M.**: Off-line geometrical and microscopic & on-line vibration based cutting tool wear analysis for micro-milling of ceramics, *Measurement*, Vol. 163., 2020., online available

[32] **Xie, J.; Wang, Y.; Zheng, X.; Yang, Q.; Wang, T.; Zou, Y.; Xing, J.; Dong, Y.**: Modeling and forecasting Acinetobacter baumannii resistance to set appropriate use of cefoperazone-sulbactam: Results from trend analysis of antimicrobial consumption and development of resistance in a tertiary care hospital, *American Journal of Infection Control*, vol.43, 2015, pp.861-864.

[33] **Gan, M.; Cheng, Y.; Liu, K.; Zhang, G.**: Seasonal and trend time series forecasting based on a quasi-linear autoregressive model, *Applied Soft Computing*, Vol. 24, 2014, pp.13-18.

[34] **Clarkson, C. R.; Williams-Kovacs, J. D.; Qanbari, F.; Behmanesh, H.; Sureshjani, M. H.**: History-matching and forecasting tight/shale gas condensate wells using combined analytical, semi-analytical, and empirical methods, *Journal of Natural Gas Science and Engineering*, Vol. 26, 2015, pp.1620-1647.

[35] **Koprinska, I.; Rana, M.; Lora, A.T.; Martínez-Álvarez, F.**: Combining pattern sequence similarity with neural networks for forecasting electricity demand time series, *The 2013 International Joint Conference on Neural Networks (IJCNN)*, 2013., pp.1-8.

[36] **Semenychev, V. K.; Kurkin, E. I.; Semenychev, E. V.**: Modelling and forecasting the trends of life cycle curves in the production of non-renewable resources, *Energy*, Vol.75., 2014, pp. 244-251.

[37] **Paggi, R.; Mariotti, G. L.; Paggi, A.; Calogero, A.; Leccese, F.**: Prognostics via Physics-Based Probabilistic Simulation Approaches, *Proc. of Metrology for Aerospace, 3rd IEEE International Workshop*, Firenze, Italy, June 21-23, 2016, pp. 130 - 135.

[38] **Ed. Francesco, De; De Francesco, Ett.; De Francesco, R.; Leccese, F.**; Cagnetti, M.: Improving Autonomic Logistic analysis by including the production compliancy status as initial degradation state, *Proc. of Metrology for Aerospace, 3rd IEEE International Workshop*, Firenze, Italy, June 21-23, 2016., pp. 371 - 375.

[39] **Sutton, R..; Barto, A. G..**: Reinforcement Learning: An Introduction, *Book, The MIT Press*, 2018.

[40] **OpenAI**. Openai five. https://blog.openai.com/openai-five/, 2018

[41] **Xu, Z.; van Hasselt, H.; Silver, D.**: Meta-gradient reinforcement learning. *arXiv*, preprint arXiv:1805.09801, 2018.

[42] **van Otterlo, M.; Wiering, M.:** Reinforcement Learning and Markov Decision Processes, *Wiering M., van Otterlo M. (eds) Reinforcement Learning. Adaptation, Learning, and Optimization*, Vol 12. Springer, Berlin, Heidelberg, 2012, https://doi.org/10.1007/978-3-642-27645-3_1

[43] **Andersen, R.E., Madsen, S., Barlo, A.B.K., Johansen, S.B., Nør, M., Andersen, R.S., Bøgh, S.**, 2019. Self-learning Processes in Smart Factories: Deep Reinforcement Learning for Process Control of Robot Brine Injection. *Procedia Manufacturing* 38, 171–177. https://doi.org/10.1016/j.promfg.2020.01.023

[44] **Beruvides, G., Villalonga, A., Franciosa, P., Ceglarek, D., Haber, R.E.**, 2018. Fault pattern identification in multi-stage assembly processes with non-ideal sheet-metal parts based on reinforcement learning architecture. *Procedia CIRP* 67, 601–606. https://doi.org/10.1016/j.procir.2017.12.268

[45] **Guo, F., Zhou, X., Liu, J., Zhang, Y., Li, D., Zhou, H**., 2019. A reinforcement learning decision model for online process parameters optimization from offline data in injection molding. *Applied Soft Computing* 85, 105828. https://doi.org/10.1016/j.asoc.2019.105828

[46] **Li, F., Chen, Y., Wang, J., Zhou, X., Tang, B.**, 2019. A reinforcement learning unit matching recurrent neural network for the state trend prediction of rolling bearings. *Measurement* 145, 191–203. https://doi.org/10.1016/j.measurement.2019.05.093

[47] **Wang, R., Jiang, H., Li, X., Liu, S.**, 2020. A reinforcement neural architecture search method for rolling bearing fault diagnosis. *Measurement* 154, 107417. https://doi.org/10.1016/j.measurement.2019.107417

[48] **Ramanathan, P., Mangla, K.K., Satpathy, S.**, 2018. Smart controller for conical tank system using reinforcement learning algorithm. *Measurement* 116, 422–428. https://doi.org/10.1016/j.measurement.2017.11.007

[49] **Wu, H.; Dai, Y.; Wang, C.; Xu, X.; Jiang, X.**, 2020, Identification and forewarning of GNSS deformation information based on a modified EWMA control chart, *Measurement*, Volume 160, 107854, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2020.107854.

[50] **Aslam, M.; Srinivasa Rao, G.; Shafqat, A.; Ahmad, L.; Sherwani,R. A. K.**, 2021, Monitoring circuit boards products in the presence of indeterminacy, *Measurement*, Volume 168, 108404, ISSN 0263-2241, https://doi.org/10.1016/j.measurement.2020.108404.

[51] **Viharos, Zs. J.; Jakab, R. B.:** Reinforcement Learning for Statistical Process Control in Manufacturing, *17th IMEKO TC 10 and EUROLAB Virtual Conference: "Global Trends in Testing, Diagnostics & Inspection for 2030"*, October 20-22., 2020., ISBN: 978-92-990084-6-1, pp. 225-234.

[52] **Kavimani, V.; Prakash, K. S.; Thankachan, T.**: Multi-objective optimization in WEDM process of graphene – SiC-magnesium composite through hybrid techniques, *Meas. J. Int. Meas. Confed.*, vol. 145, pp. 335–349, Oct. 2019, doi: 10.1016/j.measurement.2019.04.076.

[53] **Aquila, G.; Peruchi, R.S.; Rotela, P.; Rocha, Jun.L.C.S.**; de Queiroz, A.R.; de O. Pamplona, E.; Balestrass, P.P.: Analysis of the wind average speed in different Brazilian states using the nested GR&R measurement system, *Meas. J. Int. Meas. Confed.*, vol. 115, pp. 217–222, Feb. 2018, doi: 10.1016/j.measurement.2017.10.048.

[54] **Mba, C. U.; Makis, V.; Marchesiello, S.; Fasana, A.; Garibaldi, L.**: Condition monitoring and state classification of gearboxes using stochastic resonance and hidden Markov models, *Meas. J. Int. Meas. Confed.*, vol. 126, pp. 76–95, Oct. 2018, doi: 10.1016/j.measurement.2018.05.038.

[55] **Peng, G.; Zhang, Z.; Li, W.**: Computer vision algorithm for measurement and inspection of O-rings, *Meas. J. Int. Meas. Confed.*, vol. 94, pp. 828–836, Dec. 2016, doi: 10.1016/j.measurement.2016.09.012.

[56] **Wu, F.; Li, Q.; Li, S.; Wu, T.**: Train rail defect classification detection and its parameters learning method, *Meas. J. Int. Meas. Confed.*, vol. 151, p. 107246, Feb. 2020, doi: 10.1016/j.measurement.2019.107246.

[57] **Gopal, P. M.; Prakash, K.S.**: Minimization of cutting force, temperature and surface roughness through GRA, TOPSIS and Taguchi techniques in end milling of Mg hybrid MMC, *Meas. J. Int. Meas. Confed.*, vol. 116, pp. 178–192, Feb. 2018, doi: 10.1016/j.measurement.2017.11.011.

[58] **Prakash, K.S.; Gopal, P.M.; Karthik, S.**: Multi-objective optimization using Taguchi based grey relational analysis in turning of Rock dust reinforced Aluminum MMC, *Meas. J. Int. Meas. Confed.*, vol. 157, p. 107664, Jun. 2020, doi: 10.1016/j.measurement.2020.107664.

[59] **Alam, S.; Gupta, R.; Bera, J.**: Quality controlled compression technique for Photoplethysmogram monitoring applications, *Meas. J. Int. Meas. Confed.*, vol. 130, pp. 236–245, Dec. 2018, doi: 10.1016/j.measurement.2018.07.091.

[60] https://www.ipar4.hu/page/ipari-forradalmak-ipar-4-0