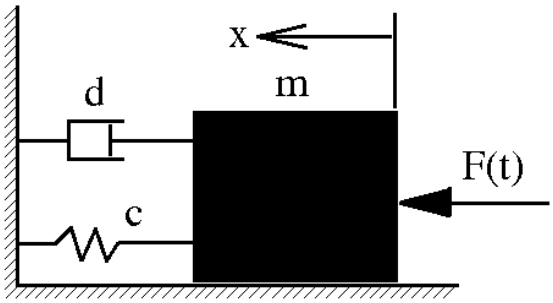
11. példa: Egy szabadságfokú lengőrendszer (program6_1.vi)

Ennek a feladatnak a megoldása során egy egy szabadságfokú lengőrendszer szimulációját valósítjuk meg. A rendszer kialakítását a VI.1. ábra mutatja, ahol a feltüntetett paraméterek a következők:

- m – a mozgást végző test tömege,
- d – a rendszer csillapítási tényezője,
- s – a rendszer rugóállandója,
- F(t) – a rendszert érő gerjesztés,
- x – a tömeg elmozdulása.



VI.1. ábra: A rendszer egy lehetséges kialakítása

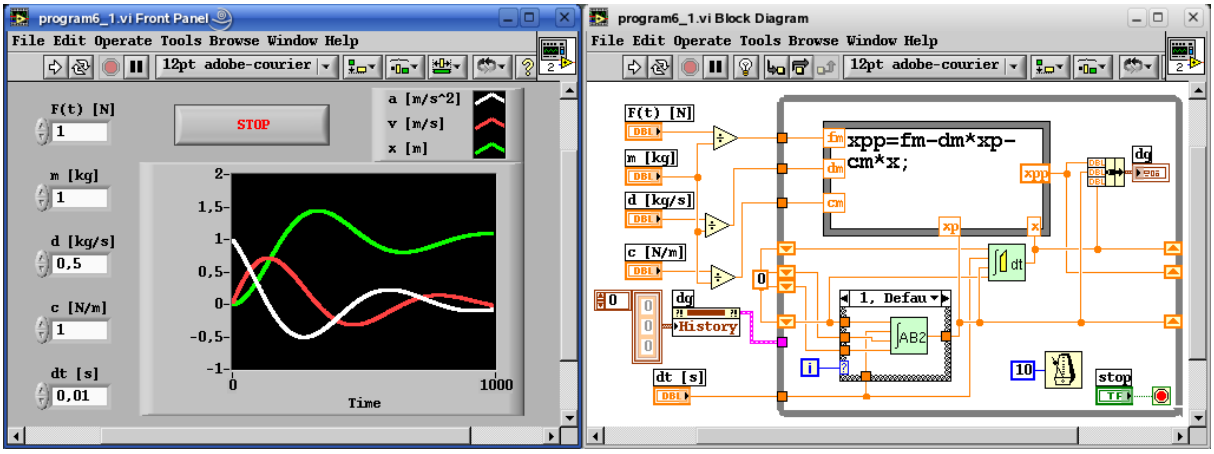
A megoldáshoz a rendszer differenciálegyenletét kell felírnunk, mely a következőképpen néz ki:

$$m \cdot \ddot{x} + d \cdot \dot{x} + c \cdot x = F(t) \tag{1}$$

A szimuláció elvégzéshez célszerű a legmagasabb fokú deriváltat kifejezni:

$$\ddot{x} = \frac{F(t)}{m} - \frac{d}{m} \cdot \dot{x} - \frac{c}{m} \cdot x \tag{2}$$

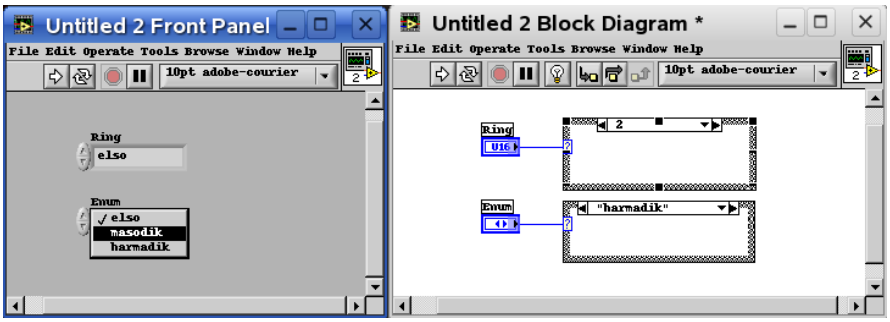
A rendszert egy másodrendű differenciálegyenlet írja le, tehát a megoldásunk során két integrátort kell majd használnunk. Ez lehet két párhuzamosan kötött másodrendű Adams-Bashforth integrátor, vagy egy másodrendű Adams-Bashforth integrátor és vele sorba kötve egy másodrendű AdamsMoulton integrátor. Mivel ez utóbbi megoldás pontosabb eredményt ad a szimuláció során, ezért ezt készítsük el!



VI.2. ábra: Az 1DOF lengőrendszert megvalósító szimulációs program (v1a)

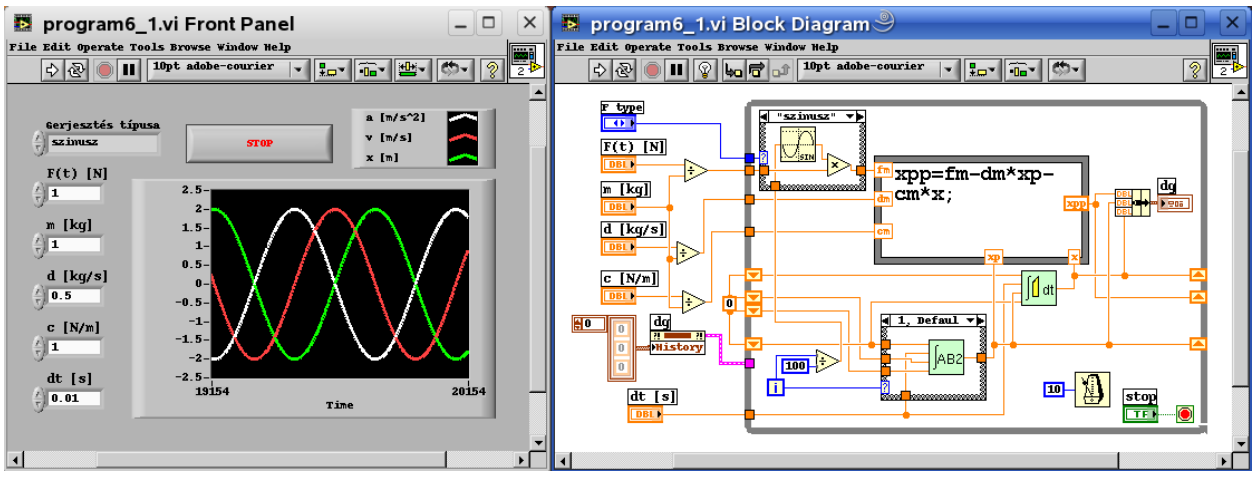
Amennyiben a gerjesztés F(t)=F típusú, akkor a feladat megoldásához, az eddig megszerzett LabVIEW ismereteink elegendőek. A VI.2. ábra egy lehetséges megvalósítást, illetve a beállított paraméterekkel történt szimuláció eredményét mutatja be.

Amennyiben több típusú gerjesztést is szeretnénk a rendszerünknek biztosítani, akkor célszerű megismerkednünk a 'Ring' illetve az 'Enum' adattípussal. Ezek tulajdonképpen egész típusú változók, amelyeknek az egyes értékeihez neveket rendelhetünk. A két típus közötti különbséget a VI.3. ábra mutatja be.



VI.3. ábra: A 'Ring' és az 'Enum' típusok közötti különbség

A programunkat tehát bővítsük ki azzal a lehetőséggel, hogy az egységugrás jellegű gerjesztés mellett szinuszos jellegű gerjesztést is lehessen benne alkalmazni! Ehhez használjunk egy 'Enum' típusú változót. A módosított program, illetve egy futtatás eredménye a VI.4. ábrán látható.

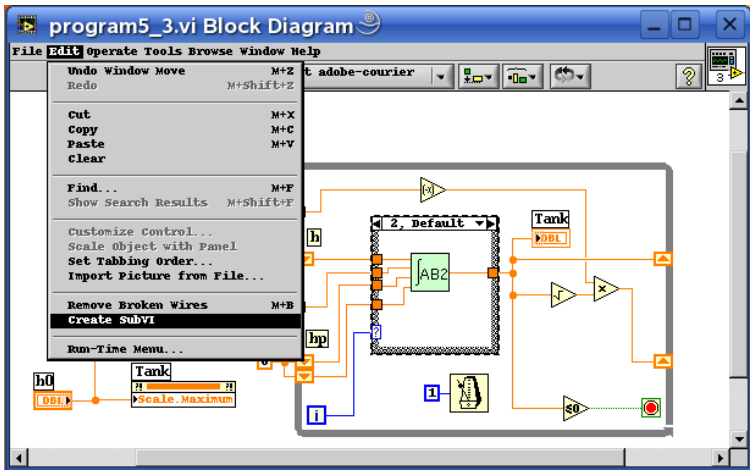


VI.4. ábra: Az 1DOF lengőrendszert megvalósító szimulációs program (v1b)

12. példa: Tartályból való kifolyás – v2 (program6_2.vi)

A következő három példa során az előző fejezetben tárgyalt tartályból való kifolyást bemutató, illetve az imént elkészített lengőrendszer programját fogjuk átírni oly módon, hogy közönséges integrátorok helyett úgynevezett hisztorikus integrátorokat alkalmazunk a programozás során.
Mik azok a hisztorikus integrátorok és miért előnyös az alkalmazásuk? Ha az egyik kérdésre válaszolunk, akkor tulajdonképpen a másikra is meg fogjuk adni a választ. Ha egy integrátor hisztorikus, akkor az azt jelenti, hogy képes „emlékezni” a korábbi időpillanatbeli integrál-valamint függvényértékekre. Ennek következtében kevesebb paramétert kell számára biztosítani,

azaz kevesebb bemenete lesz az adott integrátort megvalósító 'subVI'-nak. Ez különösen jelentős magasabb rendű integrátorok esetén.
Ezután a rövid ismertető után kezdjünk neki a feladatnak. Nyissuk meg a tartályból való kifolyást modellező programunkat és mentsük el új néven, hiszen az eredeti programot is meg akarjuk tartani.
Jelöljük ki 'Block Diagram'-on az integrátorokat tartalmazó 'Case' struktúrát, és az 'Edit' menü 'Create SubVI' pontjának segítségével készítsünk belőle 'SubVI'-t (VI.5. ábra)! Ebből a 'SubVI'-ból fogjuk elkészíteni a másodrendű hisztorikus Adams-Bashforth integrátort.

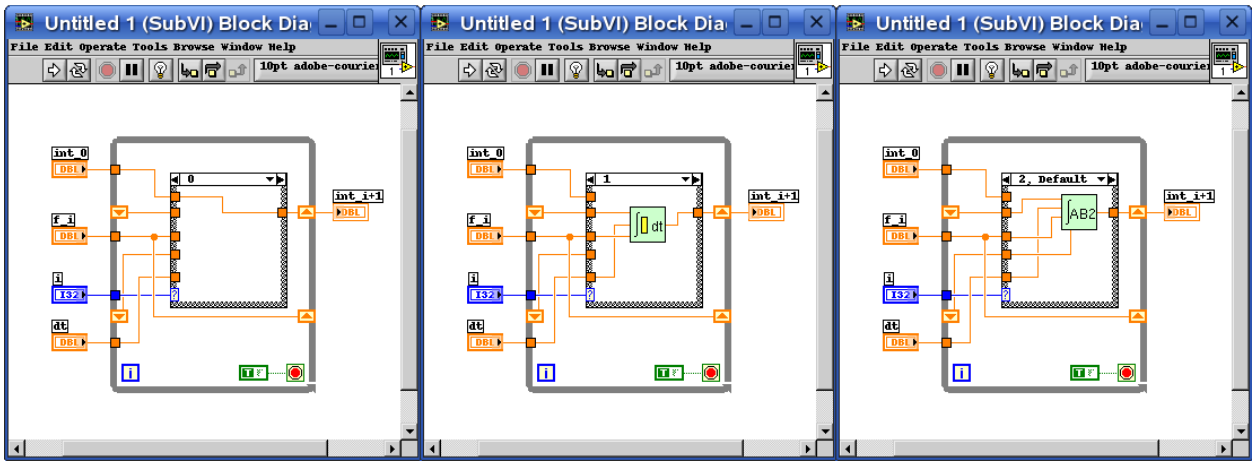


VI.5. ábra: 'SubVI' készítése kijelölésből

Jogosan vetődik fel a kérdés, hogy hogyan fogjuk elérni, hogy emlékezete legyen az integrátorunknak. Emlékezzünk vissza, amikor a 'Shift Register'-ekkel ismerkedtünk meg és nem adtunk kezdeti értéket neki, akkor a 'Shift Register' kezdeti értéke mindig az utoljára belevezetett érték volt. Már akkor utaltunk rá, hogy ezt a tulajdonságát a későbbi feladataink során ki fogjuk használni. Nos itt az alkalom!
Egy dupla kattintással nyissuk meg az új 'SubVI'-unkat! Ezután első lépésként mentsük el a VI-t, hiszen egyenlőre még csak a memóriában létezik. Javaslom, hogy az 'Integrators.llb' LabVIEW könyvtárba mentsük a többi integrátorunk közé!
Végignézve a 'Front Panel'-en láthatjuk, hogy az egyes elemeink elnevezése jelen állapotában nem tükrözi a funkcióját, úgyhogy következő lépésként nevezzük el megfelelőn őket. Ehhez célszerű az Adams-Bashforth integrátort tartalmazó esetből kiindulni, és az alapján elvégezni az átnevezéseket!
Mielőtt elhelyeznénk a 'Diagam Panel'-en egy ciklust (hiszen a 'Shift Register' csak egy ciklusban létezhet), gondoljuk át, hogy milyen lépéseket kell megvalósítanunk ebben a programban!

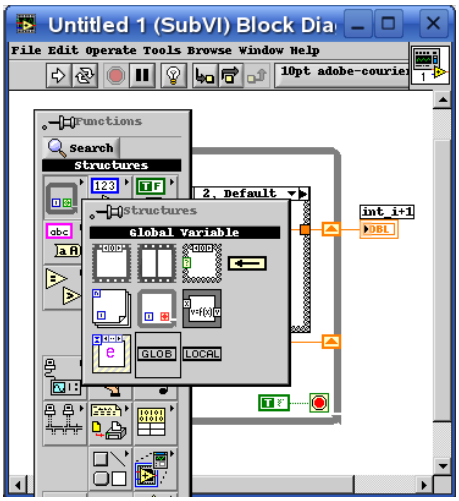
- Kezdeti értékek átvezetése – hiszen a szokásos módon nem adhatunk kezdeti értéket a 'Shift Register'-nek, mivel akkor a VI minden meghívásakor törölnénk az emlékezetét az integrátornak.
- Első integrálás téglány integrátorral.
- Integrálás másodrendű Adams-Bashforth integrátorral.

A három lépés megvalósítását mutatja a VI.6. ábra, ahol a ciklus leállási feltétele úgy van megadva, hogy az csak egyszer fusson le, hiszen a ciklusra csupán a 'Shift Register' miatt van szükségünk!



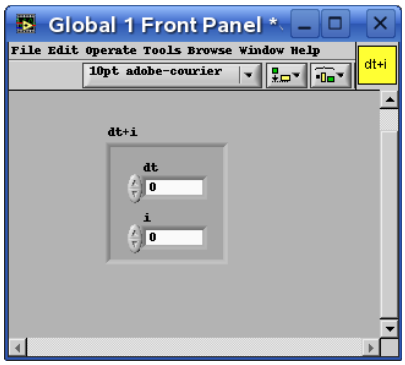
VI.6. ábra: A hisztorikus Adams-Bashforth integrátor programjának első változata

Az „i” ciklusváltozó valamint a „dt” integrálási lépésköz olyan paraméterek, amelyek – még ha több integrátort is használunk egy programban – minden integrátornál azonosak, tehát ha lenne egy olyan eszközünk, aminek a segítségével egyszerre több VI-nak is át tudjuk adni az értékeket, akkor tovább tudjuk csökkenteni a bemenetek számát. Szerencsére van ilyen eszköz a LabVIEW-ban, méghozzá az úgynevezett globális változó (VI.7. ábra).



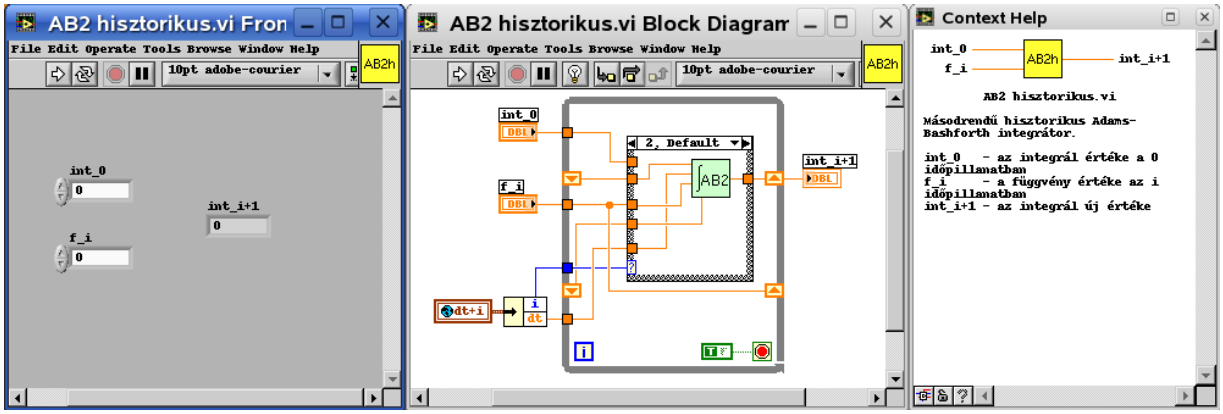
VI.7. ábra: A globális változó

Amikor a globális változót elhelyezzük a 'Diagram Panel'-en, akkor fekete színű, tehát definiálatlan adattípusú. Ahhoz, hogy ezen változtassunk, kattintsunk rá duplán, mire megjelenik a globális változó 'Front Panel'-je. Ezen tudjuk kialakítani a globális változó struktúráját. A logikailag összetartozó adatokat célszerű 'Cluster'-ben elhelyezni. A VI.8. ábrának megfelelően készítsük el mi is a ciklusváltozó és a lépésköz tárolására alkalmas struktúrát!
A globális változót felfoghatjuk úgy is, mint egy olyan adatok tárolására alkalmas 'SubVI'-t aminek nincs 'Diagram Panel'-je. Ennek megfelelően el is kell mentenünk mielőtt bezárnánk (tegyük ezt úgy, hogy az Integrators.llb-be mentjük). Ha bezártuk, még nincs hozzárendelve az új struktúra a hisztorikus integrátorunk 'Block Diagram'-ján lévő globális változóhoz. Ezt úgy tehetjük meg, hogy a változóra kattintunk a működtető típusú kurzorral ( ), és a felkínált (jelen esetben egyetlen) lehetőségek közül kiválasztjuk a megfelelőt.



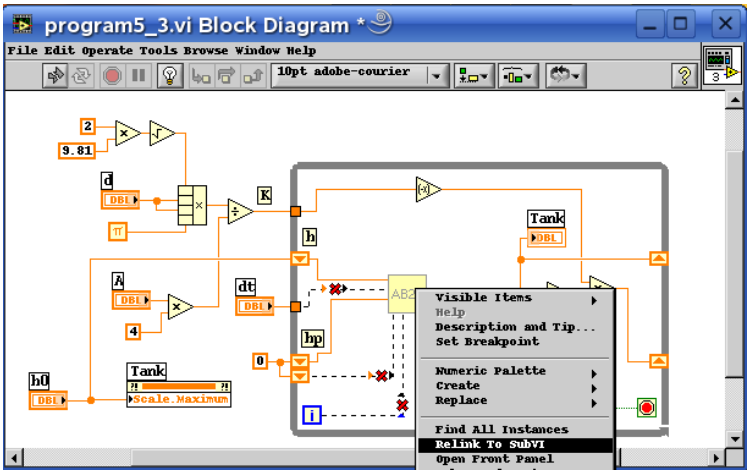
VI.8. ábra: A "dt+i" globális változó 'Front Panel'-je

A globális változó alkalmazásával immár csak két bemenete van a hisztorikus integrátornak. Mivel jelentősen lecsökkent a kapcsolódási pontok száma, ezért változtassuk meg a VI csatlakozókiosztását egy 2+1 típusúra (Az 'Icon Connector'-on jobb klikk, és a menüben a 'Patterns' pont)! Az elkészült 'SubVI'-t a VI.9. ábra mutatja.



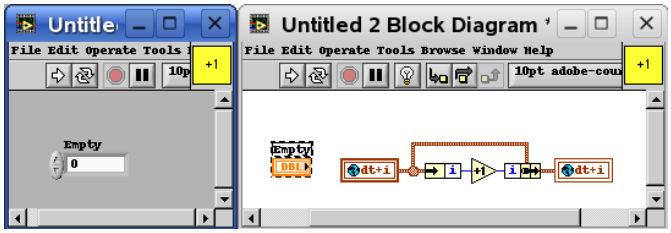
VI.9. ábra: A hisztorikus Adams-Bashforth integrátor

Visszatérvén a tartályos programunkhoz, azt tapasztaljuk, hogy az integrátor VI-a szürkés színű, és természetesen vezet felé néhány, immár felesleges és ebből kifolyólag sehova sem csatlakozó adatvezeték.



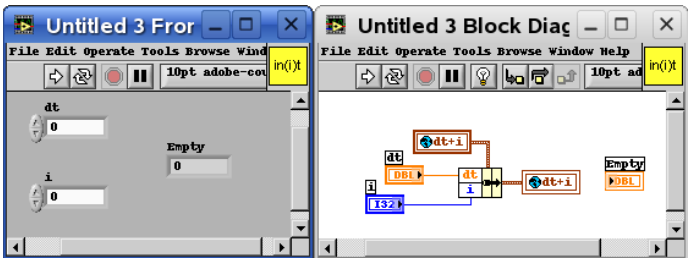
VI.10. ábra: 'SubVI' újracsatlakoztatása

Amennyiben szerencsénk van, akkor pont azok a vezetékek „törtek el”, amelyeket nem használunk a továbbiakban, és akkor ezeket egyszerűen a :Ctrl-B / :Alt-B billentyűkombinációval eltűntethetjük. A 'SubVI'-t a megváltoztatott csatlakozókiosztás miatt pedig újra kell linkelni. Ezt a hozzá tartozó legördülő menüből a 'Relink To SubVI' pontot kiválasztva tudjuk elvégezni (VI.10. ábra).
Sajnos azonban nem annyira egyszerű a feladat, hogy máris befejeztük volna. Két dolgot kell még biztosítani. Az egyik a globális változóban a ciklusváltozó növelése minden egyes integrálás után. Ehhez készítsük el a VI.11. ábrán látható VI-t! Az 'Empty' nevű bemenet arra szolgál, hogy ezen keresztül beköthessük a VI az adatfolyamba, miáltal el tudjuk érni, hogy ne kelljen sorrendi struktúrával előírni, hogy ez a 'SubVI' csak az integrálás megtörténte után hajtódjon végre.



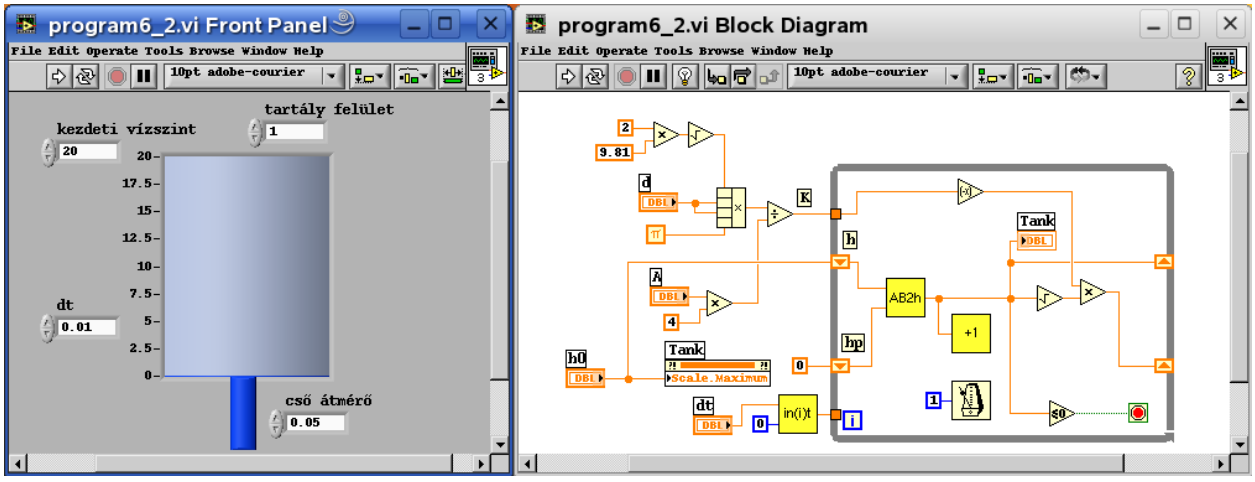
VI.11. ábra: Ciklusváltozó növelése 'subVI'

A másik a feladat elején a globális változó inicializálása (VI.12. ábra). (Az 'Empty' nevű kimenet célja teljesen hasonló mint az előző VI-nál.)



VI.12. ábra: Globális változó inicializálása

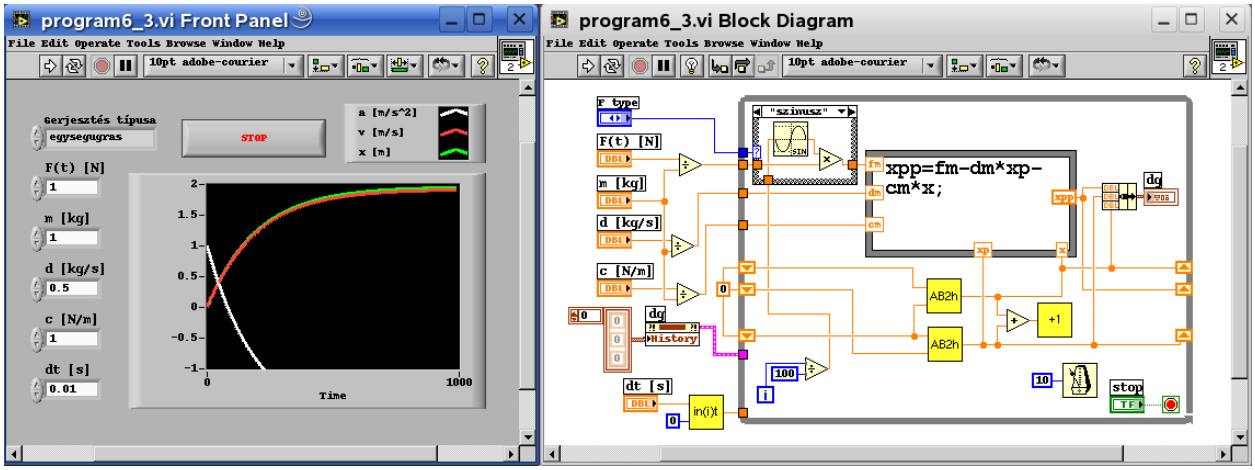
Az immár hisztorikus integrátorral megvalósított program a VI.13. ábrán látható.



VI.13. ábra: A tartályból kifolyás program hisztorikus integrátorral

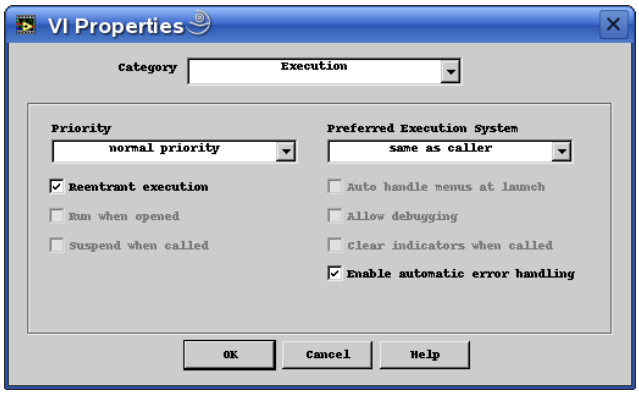
13. példa: Egy szabadságfokú lengőrendszer – v2 (program6_3.vi)

Készítsük el a lengőrendszert modellező program hisztorikus változatát is! Azonban vigyázzunk, hiszen jelenleg csak Adams-Bashforth integrátorból készítettünk csak hisztorikusat, így ezeket párhuzamosan kell bekötnünk! Az átalakított program és a futtatásának eredménye a VI.14. ábrán látható.



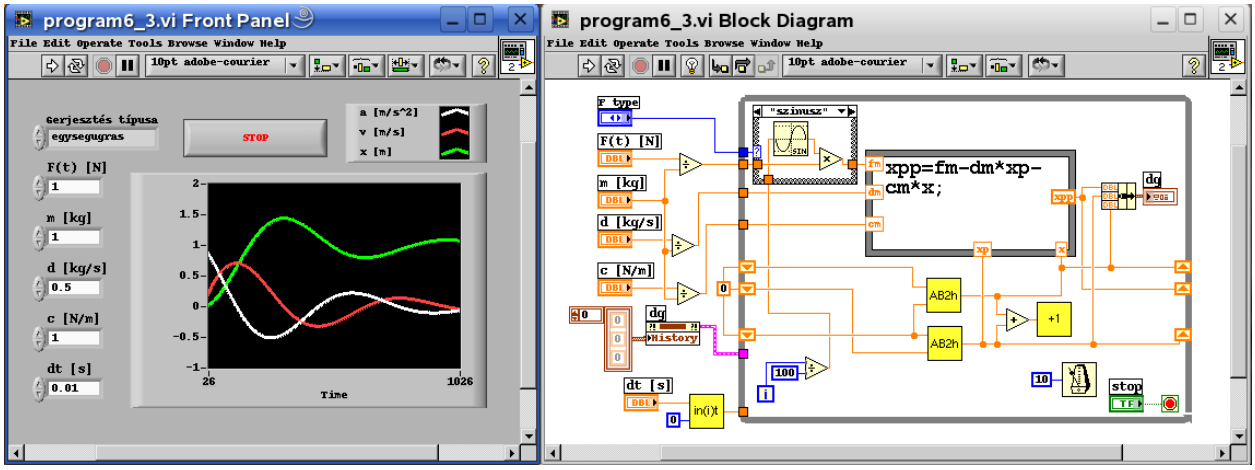
VI.14. ábra: A lengőrendszer megvalósítása párhuzamosan kapcsolt AB2h integrátorokkal (hibás működés)

Azonban, ha visszalapozunk a fejezet elejére és összehasonlítjuk a jelenlegi eredményt az akkorival, akkor kiderül, hogy valahol hiba van a programban.
Ez a hiba abból adódik, hogy két azonos hisztorikus integrátorunk van a programban és normál esetben egy 'SubVI'-nak – még ha az többször is szerepel a programban – a LabVIEW egy memóriaterületet foglal le. Azaz ebben az esetben a két hisztorikus integrátorunk 'Shift Register'-ei felülírták egymást.
Ezen úgy tudunk változtatni, hogy megnyitjuk az integrátor VI-át, és a 'VI properties'-nél az 'Execution' kategóriában bepipáljuk a 'Reentrant execution' pontot (VI.15. ábra), amely azt fogja eredményezni, hogy a ezután ennek a VI-nak minden egyes példányához külön memóriaterületet fog lefoglalni a LabVIEW.



VI.15. ábra: A 'Reentrant' működés beállítása

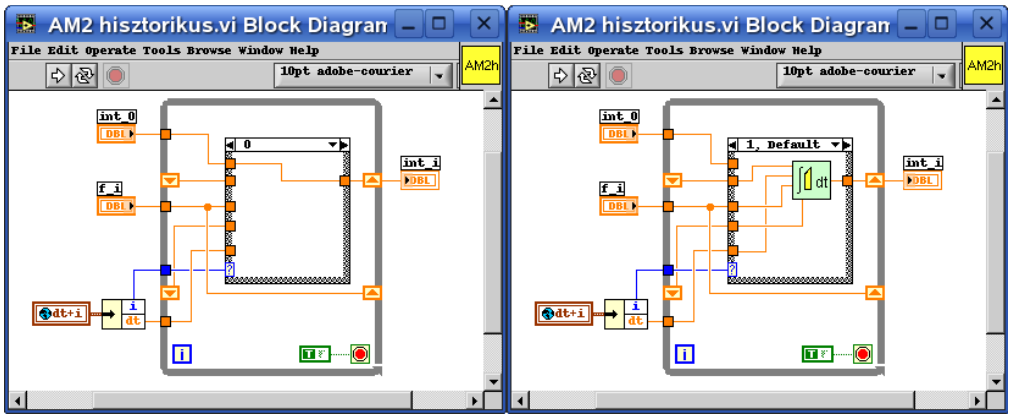
Az így módosított integrátorokkal már helyes eredményt fogunk kapni, ahogy azt a VI.16. ábra is mutatja.



VI.16. ábra: A lengőrendszer megvalósítása párhuzamosan kapcsolt AB2h integrátorokkal

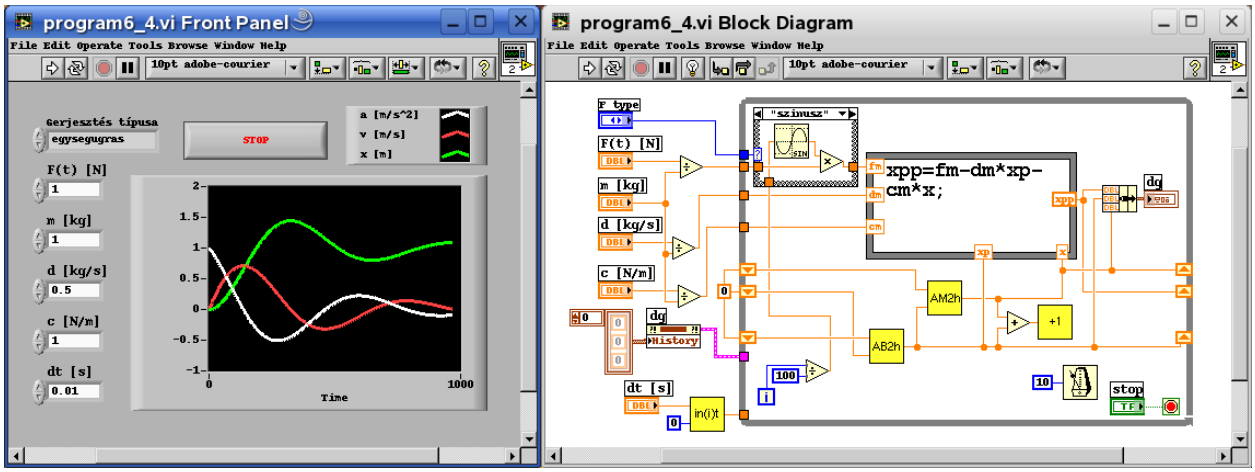
14. példa: Egy szabadságfokú lengőrendszer – v3 (program6_4.vi)

Ahogy azt az előző példában is láttuk a magasabb rendű differenciálegyenletek megoldhatók párhuzamosan kapcsolt Adams-Bashforth integrátorokkal. De ennek az integrátortípusnak relatíve nagy a hibája, így jobban járunk, ha csak a szükséges egy darabot használjuk belőle, és a többi integrátor pedig vele sorba kötve, Adams-Moulton típusú.
Készítsük tehát el a másodrendű hisztorikus Adams-Moulton integrátorunkat is. Ez némileg egyszerűbb kialakítású lesz, hiszen itt a kezdeti értékek bevitele után már csak az Adams-Moulton integrátorral való számítási műveleteket kel csak megvalósítani (VI.17. ábra).



VI.17. ábra: Másodrendű Adams-Moulton (trapéz) hisztorikus integrátor

Ne felejtsük el ennél az integrátornál sem beállítani a 'Reentrant' típusú működést, hiszen egy harmad, vagy magasabb fokú differenciálegyenlet megoldásánál ebből a típusból is többet kell felhasználni a megoldás során.
A VI.18. ábra mutatja az elkészített program 'Front Panel'-jét és 'Block Diagram'-ját, valamint egy futtatás eredményét.



VI.18. ábra: A lengőrendszer megvalósítása sorosan kapcsolt AB2h és AM2h integrátorokkal